



Simon Walz

# Ein System zur Gerätevernetzung für das Energiemanagement



Technische  
Universität  
Braunschweig



Institut für Nachrichtentechnik



# **Ein System zur Gerätevernetzung für das Energiemanagement**

Von der Fakultät für Elektrotechnik, Informationstechnik, Physik  
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades eines Doktors  
der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von Simon Walz  
aus Kassel

eingereicht am: 29.07.2019

mündliche Prüfung am: 28.11.2019

1. Referent: Prof. Dr.-Ing. Ulrich Reimers

2. Referent: Prof. Dr.-Ing. Lars Wolf

Prüfungsvorsitzender: Prof. Dr.-Ing. Thomas Kürner

Druckjahr: 2020

Dissertation an der Technischen Universität Braunschweig,  
Fakultät für Elektrotechnik, Informationstechnik, Physik

Mitteilungen aus dem Institut für Nachrichtentechnik der  
Technischen Universität Braunschweig

Band 62

**Simon Walz**

**Ein System zur Gerätevernetzung für  
das Energiemanagement**

Shaker Verlag  
Düren 2020

**Bibliografische Information der Deutschen Nationalbibliothek**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

Zugl.: Braunschweig, Techn. Univ., Diss., 2019

Copyright Shaker Verlag 2020

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungsanlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 978-3-8440-7360-7

ISSN 1865-2484

Shaker Verlag GmbH • Am Langen Graben 15a • 52353 Düren

Telefon: 02421 / 99 0 11 - 0 • Telefax: 02421 / 99 0 11 - 9

Internet: [www.shaker.de](http://www.shaker.de) • E-Mail: [info@shaker.de](mailto:info@shaker.de)

# Vorwort

Diese Arbeit ist im Rahmen meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Nachrichtentechnik der Technischen Universität Braunschweig entstanden. Da mich das Thema der Gerätevernetzung seit vielen Jahren begeistert, geht mit dieser Dissertation ein lang verfolgtes Ziel in Erfüllung, eine Lösung aktueller Problematiken für das Energiemanagement – aber auch darüber hinaus – präsentieren zu können. Die folgenden Zeilen möchte ich nutzen, um mich für die Unterstützung während der Zeit dieser Ausarbeitung zu bedanken.

An erster Stelle gilt mein Dank meinem Doktorvater Herrn Prof. Dr.-Ing. Ulrich Reimers für die fachliche Anleitung, für das Vertrauen und die Geduld sowie den ständigen Dialog auch in schwierigen Phasen. Die wertvollen Ratschläge verbunden mit den optimalen Rahmenbedingungen haben zum Erfolg dieser Arbeit maßgeblich beigetragen!

Außerdem gilt mein Dank Herrn Prof. Dr.-Ing. Lars Wolf für die Übernahme des Zweitgutachtens und Herrn Prof. Dr.-Ing. Thomas Kürner für den Vorsitz der Prüfungskommission.

Darüberhinaus gilt mein Dank allen Professoren sowie ehemaligen und aktuellen Kolleginnen und Kollegen des Instituts für Nachrichtentechnik für die vielfältige Unterstützung und die angenehme Arbeitsatmosphäre. Viele von euch sind für mich zu engen Freunden geworden.

Für die gute Zusammenarbeit möchte ich auch den Professorinnen und Professoren sowie den Kolleginnen und Kollegen des Instituts für Betriebssysteme und Rechnerverbund, des Instituts für Werkzeugmaschinen und Fertigungstechnik, des Instituts für Gebäude und Solartechnik, des Instituts für Hochspannungstechnik und Elektrische Energieanlagen sowie vieler weiterer Institute und Einrichtungen der Technischen Universität Braunschweig danken.

Des Weiteren bedanke ich mich bei den Studierenden, die im Rahmen ihrer Abschlussarbeiten die gemeinsam bearbeiteten Themen und Ideen mit Leben gefüllt haben.

Euch, liebe Freunde, liebe Mittänzerinnen und -tänzer, liebe Mitglieder der ags – Wissenschaftlichen Arbeitsgemeinschaft für Studio- und Senderfragen an der Technischen Universität Braunschweig e.V. möchte ich besonders für die mentale Unterstützung, den starken Zusammenhalt und euer Verständnis danken.

Abschließend möchte ich mich von Herzen bei meinen Eltern und meinem Bruder für den Rückhalt, die stetige Ermutigung und die umfangreiche Unterstützung bedanken.

Danke!

Braunschweig im Februar 2020

Simon Walz



## Kurzfassung

Im Zuge der für Deutschland beschlossenen Energiewende lassen sich mit Energiemanagementlösungen wertvolle Beiträge zur Erhöhung der Energieeffizienz erzielen. Mit geeigneten Energiemanagementsystemen kann die Reduzierung des Ausstoßes an Treibhausgasen bewirkt sowie eine Energiekostenreduzierung für Bürgerinnen und Bürger realisiert werden. Im Rahmen dieser Arbeit werden die wesentlichen Bestandteile von Energiemanagementsystemen und deren Einsatzfähigkeit in verteilten Systemen genauer betrachtet.

Dabei wird gezeigt, dass viele bisherige Energiemanagementsysteme Beschränkungen in ihrer Einsatzfähigkeit unterliegen. Aus diesem Grund wird ein Systemkonzept für zukünftige Energiemanagementsysteme auf Basis des Internet der Dinge (engl. Internet of Things, IoT) nach einem allgemeinen, Nutzer-zentrierten Ansatz entwickelt, welches die vorhandenen Restriktionen eliminieren soll. Das Systemkonzept besteht aus drei Schichten. Für die untere Schicht werden Verfahren zur Geräteeinbindung dargelegt, mit denen möglichst alle Arten von Anlagen, Maschinen und Geräten eingebunden werden können. Als Übergangslösung für bestehende Geräte ohne (geeignete) Daten-Schnittstellen sollen deren fehlende Daten und Funktionen mit Hilfe der Signalverarbeitung und -erkennung nachgebildet werden. In der mittleren Schicht sollen Daten möglichst lokal vermittelt und verarbeitet werden, um Anforderungen an den Datenschutz und die Datensicherheit einhalten zu können. Hierzu wird eine hierarchische Kommunikationsstruktur unter Verwendung von verteilten Anwendungen vorgeschlagen, die eine Kooperation aller Elemente ermöglicht. Für die obere Schicht werden verschiedene Energiemanagement-Funktionalitäten für den Einsatz in der hierarchischen Kommunikationsstruktur entworfen.

Für die Erprobung des Systemkonzepts wurden verschiedene Implementierungen erstellt: Eine Softwareplattform in der Skriptsprache JavaScript erlaubt die Ausführung von verteilten Anwendungen. Verschiedene Techniken zur Geräteeinbindung sowie Energiemanagement-Funktionalitäten wurden als Anwendungen der Softwareplattform implementiert. Dazu zählen eine Energieverbrauchsschätzung für Computer, mehrere Techniken zur Identifikation des Gerätebetriebsstatus anhand von Messdaten, einige Smartphone- bzw. Tablet-Anwendungen für die Nutzer-unterstützte Energieverbrauchsoptimierung sowie Anwendungen zur automatisierten Energieverbrauchsoptimierung inklusive einer Lastspitzenreduktion.

Zur Demonstration der Funktionsfähigkeit wurde das System in Unternehmen und Privathaushalten eingesetzt, wie z. B. in einer Werkzeugmaschinen-Halle, in mehreren Büroräumen, in Forschungslaboren sowie in Privaträumen. Dabei wird gezeigt, dass die bisherigen Beschränkungen der Einsatzfähigkeit von Energiemanagementsystemen überwunden werden und ein großflächiger Einsatz technisch möglich ist. Weiterhin werden die verschiedenen Komponenten und Verfahren einzeln evaluiert. Die gewählte hierarchische Kommunikationsstruktur bestätigt sich dabei als geeignetes Konzept, das zudem weitere Vorteile hinsichtlich der Skalierungsfähigkeit und der Ausfallsicherheit bietet. Die entstandene Softwareplattform eignet sich aufgrund ihrer nachgewiesenen Leistungsfähigkeit auch für einen produktiven Einsatz außerhalb der Forschung. Zusätzlich wird gezeigt, dass die verschiedenen Techniken und Funktionalitäten aufgrund ihrer Präzision und aufgrund ihrer gezeigten Realisierbarkeit angemessene Lösungen für das Energiemanagement darstellen.

# Abstract

In view of the energy transition in Germany, potential power savings are achieved through the application of energy management solutions. Using appropriate concepts, energy management systems can contribute to the reduction in the emission of greenhouse gases and reduce the energy costs for end consumers. In this thesis, the different components of energy management systems are studied and their applicability in distributed systems is analyzed.

It is shown that a majority of energy management systems suffer from limitations in their operational capabilities. For this reason, a user oriented system concept for future energy management systems based on the Internet of Things (IoT) is developed, eliminating the limitations of existing systems. The system is composed of three layers, the lowest of which addresses the methods for the integration of all types of machines and devices. As a transitional solution for existing devices without (appropriate) data interfaces, missing data and functions will be emulated using signal processing and recognition. The middle layer addresses the necessity to communicate and process data locally, if possible, to satisfy privacy and data security requirements. To that end, a hierarchical communication structure with distributed applications is suggested, which still enables the cooperation of all elements. In the top layer, different energy management functionalities for use in the hierarchical communication structure are outlined.

For the testing of the system concept, several implementations are created. A software platform in JavaScript allows for the running of distributed applications. Several device connectivity techniques as well as energy management functionalities are implemented as applications of the software platform, such as an energy consumption estimation for computers, several device status identification techniques based on measured data, several smartphone and tablet applications for the end-user based energy consumption optimization as well as some applications for the automated energy consumption optimization, including a peak consumption reduction.

To demonstrate the functionality of the system, it was implemented in both work and home environments, including a machine workshop, in several office spaces, in research labs and in private residences. Thereby it is demonstrated that the restrictions in the

operational capability of energy management systems could be overcome and that a wide deployment of these systems is possible from a technical point of view. Furthermore, the different components and processes are individually evaluated, whereby the chosen hierarchical communication structure is proven to be a practical concept, offering advantages in scaling and reliability. The resulting software platform is also suitable for a productive deployment outside research, due to its demonstrated capability. It is shown that the different techniques and functionalities offer a capable energy management solution, as a result of their precision and demonstrated feasibility.

# Inhaltsverzeichnis

|   |            |
|---|------------|
| <b>Vorwort</b>  | <b>v</b>   |
| <b>Kurzfassung</b>  | <b>vii</b> |
| <b>Abstract</b>   | <b>ix</b>  |
| <b>1 Einleitung</b>   | <b>1</b>   |
| <b>2 Stand der Technik</b>  | <b>7</b>   |
| 2.1 Energiemanagement . . . . .   | 7          |
| 2.1.1 Grundprinzipien der elektrischen Energieversorgung . . . . .              | 8          |
| 2.1.2 Beweggründe für das Energiemanagement . . . . .                           | 11         |
| 2.1.3 Beweggründe aus Sicht der Endverbraucher . . . . .                        | 15         |
| 2.1.4 Maßnahmen des Energiemanagements . . . . .                                | 16         |
| 2.2 Energiemanagementsysteme und ihre Rahmenbedingungen . . . . .               | 19         |
| 2.2.1 Automatisierte Energiemanagementsysteme für Privathaushalte . . . . .     | 21         |
| 2.2.2 Algorithmen zur automatisierten Energieverbrauchsoptimierung . . . . .    | 23         |
| 2.2.3 Energie-Informationssysteme . . . . .                                     | 26         |
| 2.2.4 Energiemanagementsysteme für Unternehmen und Einrichtungen . . . . .      | 27         |
| 2.3 Kommunikationsstrukturen für das Internet der Dinge . . . . .               | 29         |
| 2.3.1 Verwandte Infrastrukturen und Systeme . . . . .                           | 31         |
| 2.3.2 Ansätze für Kommunikationsstrukturen im Internet der Dinge . . . . .      | 32         |
| 2.3.3 Middleware für die Anbindung verteilter Geräte . . . . .                  | 37         |
| 2.3.4 Protokolle für das Internet der Dinge . . . . .                           | 38         |
| 2.3.5 IoT- und Cloud-basierte Energiemanagementlösungen . . . . .               | 40         |
| <b>3 Ansätze zur Erweiterung von Energiemanagementsystemen</b>                  | <b>43</b>  |
| 3.1 Eine Vision für Energiemanagementsysteme . . . . .                          | 44         |
| 3.2 Analyse aus der Literatur bekannter und am Markt vorhandener Lösungsansätze | 46         |
| 3.3 Ein Zielsystem zur Realisierung der Vision . . . . .                        | 49         |
| 3.3.1 Zielfunktionalitäten . . . . .  | 51         |
| 3.3.2 Ziele für die Geräteeinbindung . . . . .                                  | 53         |

|          |  |           |
|----------|--|-----------|
| 3.3.3    | Nichtfunktionale Anforderungen . . . . .   | 54        |
| 3.3.4    | Funktionale Anforderungen . . . . .  | 56        |
| <b>4</b> | <b>Systemkonzept zum Energiemanagement in verteilten Systemen</b>                                    | <b>59</b> |
| 4.1      | Einbindung von vernetzten Geräten . . . . .  | 61        |
| 4.1.1    | Notwendiger Funktionsumfang von vernetzten Geräten . . . . .   | 62        |
| 4.1.2    | Einbindung von Geräten über vorhandene Kommunikationstechniken                                       | 63        |
| 4.1.3    | Gerätemodellierung zur Nachbildung von Geräten . . . . .   | 65        |
| 4.1.4    | Nachbildung fehlender Sensordaten . . . . .  | 66        |
| 4.1.5    | Beeinflussbarkeit des Betriebsstatus von vernetzten Geräten . . . . .                                | 67        |
| 4.1.6    | Nachbildung der Funktionalität zur Beeinflussung des Betriebsstatus                                  | 68        |
| 4.2      | Einbindung von nicht-vernetzten Geräten . . . . .  | 69        |
| 4.2.1    | Einbindung als IoT-Gerätetrepräsentation . . . . .   | 71        |
| 4.2.2    | Einbindung mit Hilfe zusätzlicher Sensoren . . . . .   | 72        |
| 4.2.3    | Zustandsidentifikation mittels regelbasierter Verfahren . . . . .                                    | 74        |
| 4.2.4    | Zustandsidentifikation mittels statistischer Verfahren . . . . .                                     | 77        |
| 4.2.5    | Zustandsidentifikation mehrerer Geräte . . . . .   | 78        |
| 4.2.6    | Bereitstellung von Modellen . . . . .  | 80        |
| 4.3      | Kommunikationsstruktur für Energiemanagement-Systeme . . . . .                                       | 80        |
| 4.3.1    | Datenobjekte und deren Austausch . . . . .   | 81        |
| 4.3.2    | Hierarchischer Kommunikationsansatz . . . . .  | 82        |
| 4.3.3    | Wahrung des Datenschutzes und der Datensicherheit . . . . .  | 84        |
| 4.3.4    | Verteilte IoT-Anwendungen . . . . .  | 84        |
| 4.3.5    | Einheitliche Softwareplattform . . . . .   | 87        |
| 4.4      | Nutzerunterstützung für das Energiemanagement . . . . .  | 88        |
| 4.4.1    | Nutzerunterstützung für Expertinnen und Experten sowie Eigentü-<br>merinnen und Eigentümer . . . . . | 88        |
| 4.4.2    | Nutzerunterstützung für Anwenderinnen und Anwender . . . . .   | 89        |
| 4.4.3    | Navigationsstruktur für Nutzer-unterstützende Energiemanagement-<br>Anwendungen . . . . .            | 90        |
| 4.5      | Automatisiertes Energiemanagement . . . . .  | 91        |
| 4.5.1    | Steuerung auf Basis einer direkten Verknüpfung . . . . .   | 91        |
| 4.5.2    | Steuerung auf Basis von Schwellenwerten . . . . .  | 92        |
| 4.5.3    | Abstimmung von Geräten untereinander anhand von Optimierungsal-<br>gorithmen . . . . .               | 93        |
| 4.6      | Weitere Funktionalitäten jenseits des Energiemanagements . . . . .                                   | 94        |
| 4.6.1    | Smart Home mit Visualisierung und Steuerung von Geräten . . . . .                                    | 95        |

|          |  |            |
|----------|--|------------|
| 4.6.2    | Prädiktive Instandhaltung . . . . .  | 95         |
| 4.6.3    | Sicherheitsabschaltungen . . . . .   | 96         |
| <b>5</b> | <b>Realisierung eines Systems zum Energiemanagement</b>  | <b>97</b>  |
| 5.1      | Umsetzung der IoT-Softwareplattform . . . . .  | 98         |
| 5.1.1    | Datenressourcen: Austausch und Speicherung von Informationen . . .   | 100        |
| 5.1.2    | Basismethoden zur Interaktion mit Datenressourcen . . . . .  | 102        |
| 5.1.3    | Schnittstelle für Software-Anwendungen . . . . .   | 104        |
| 5.1.4    | Netzwerkcommunication . . . . .  | 105        |
| 5.1.5    | Zugriffsschutz: Begrenzung der Weiterleitung von Datenressourcen . .   | 107        |
| 5.1.6    | Darstellung von Nutzerschnittstellen . . . . .   | 108        |
| 5.1.7    | Implementierte Anwendungen . . . . .   | 109        |
| 5.1.8    | Geräteeinbindung von vernetzten Geräten . . . . .  | 111        |
| 5.1.9    | Geräteeinbindung von nicht-vernetzten Geräten . . . . .  | 112        |
| 5.2      | Signalverarbeitung für die Geräteeinbindung . . . . .  | 113        |
| 5.2.1    | Direkte Datenverarbeitung . . . . .  | 115        |
| 5.2.2    | Energieverbrauchsermittlung aus Statusdaten . . . . .  | 116        |
| 5.2.3    | Direkte Zuordnung des Energieverbrauchs . . . . .  | 117        |
| 5.2.4    | Energieverbrauchsermittlung von Desktop-Computern . . . . .  | 118        |
| 5.2.5    | Zustandsidentifikation aus zusätzlichen Messdaten . . . . .  | 120        |
| 5.2.6    | Zustandsidentifikation mittels Schwellenwert-Klassifikation . . . . .  | 121        |
| 5.2.7    | Modell für die Zustandsidentifikation mittels Schwellenwert-Klassifikation . . . . .                                   | 122        |
| 5.2.8    | Zustandsidentifikation anhand des DTW-Musterabgleichs . . . . .  | 124        |
| 5.2.9    | Mehrgeräteerkennung mittels künstlicher neuronaler Netze anhand von Netzspannungsbeeinflussungen . . . . .             | 126        |
| 5.3      | Exemplarische Anwendungen zum Energiemanagement . . . . .  | 128        |
| 5.3.1    | Nutzer-unterstützende Funktionalitäten zum Einsparen von Energie .   | 128        |
| 5.3.2    | Automatisiertes Energiemanagement mittels direkter Verknüpfung . .   | 132        |
| 5.3.3    | Steuerung auf Basis von Schwellenwerten . . . . .  | 135        |
| 5.3.4    | Abstimmung von Geräten untereinander anhand von Optimierungsalgorithmen . . . . .                                      | 136        |
| 5.3.5    | Funktionalitäten jenseits des Energiemanagements . . . . .   | 138        |
| <b>6</b> | <b>Einsatz und Evaluation des Systems in realen Umgebungen</b>   | <b>141</b> |
| 6.1      | Analyse der Funktionsfähigkeit des Systemkonzepts anhand des Gesamtdemonstrators „Real-Life-Lab Campus 2020“ . . . . . | 142        |
| 6.1.1    | Gesamtaufbau und Kommunikationsstruktur . . . . .  | 142        |

|          |   |            |
|----------|---|------------|
| 6.1.2    | Charakterisierung der Versuchsumgebungen und der sich ergebenden Herausforderungen . . . . .                      | 144        |
| 6.1.3    | Realisierte Energiemanagement-Anwendungen . . . . .   | 146        |
| 6.1.4    | Fazit . . . . .   | 148        |
| 6.2      | Analyse des Designs und Verhaltens der IoT-Softwareplattform . . . . .  | 149        |
| 6.2.1    | Evaluation des Systemverhaltens . . . . .   | 150        |
| 6.2.2    | Systemverhalten unter Verwendung der hierarchischen Kommunikationsstruktur . . . . .                              | 154        |
| 6.2.3    | Umsetzung der funktionalen Anforderungen . . . . .  | 157        |
| 6.2.4    | Umsetzung der nichtfunktionalen Anforderungen . . . . .   | 161        |
| 6.2.5    | Fazit . . . . .   | 166        |
| 6.3      | Untersuchung der Verfahren zur Einbindung von Geräten . . . . .   | 168        |
| 6.3.1    | Evaluationsumgebung & Hilfsanwendungen . . . . .  | 170        |
| 6.3.2    | Energieverbrauchsermittlung von Desktop-Computern . . . . .   | 171        |
| 6.3.3    | Zustandsidentifikation mittels Schwellenwert-Klassifikation anhand des Modells aus Unterabschnitt 5.2.7 . . . . . | 172        |
| 6.3.4    | Zustandsidentifikation anhand des DTW-Musterabgleichs . . . . .   | 175        |
| 6.3.5    | Mehrgeräteerkennung mittels künstlicher neuronaler Netze anhand von Netzspannungsbeeinflussungen . . . . .        | 176        |
| 6.3.6    | Fazit . . . . .   | 180        |
| 6.4      | Überprüfung der Zielfunktionalitäten . . . . .  | 182        |
| 6.5      | Diskussion und Fazit zum Systemkonzept . . . . .  | 183        |
| <b>7</b> | <b>Zusammenfassung und Ausblick</b>   | <b>191</b> |
|          | <b>Begriffs- und Abkürzungsverzeichnis</b>  | <b>197</b> |
|          | <b>Symbolverzeichnis</b>  | <b>202</b> |
|          | <b>Abbildungsverzeichnis</b>  | <b>205</b> |
|          | <b>Verzeichnis der eigenen Veröffentlichungen</b>   | <b>211</b> |
|          | <b>Verzeichnis der betreuten studentischen Arbeiten</b>   | <b>212</b> |
|          | <b>Verzeichnis der referenzierten Gesetze</b>   | <b>214</b> |
|          | <b>Literaturverzeichnis</b>   | <b>216</b> |



# 1 Einleitung

Viele unserer täglichen technischen Gebrauchsgegenstände erfüllen ihre Funktion nur durch den *Einsatz von Energie*. Durch die Umwandlung von elektrischer Energie oder fossiler Energiequellen in thermische Energie, wie z. B. Wärme und Kälte, oder mechanische Energie optimieren wir unsere täglichen Lebensumgebungen. Anlagen, Maschinen und Geräte überführen dazu Energie in andere Energieformen, um damit eine Aufgabe zu erfüllen (vgl. [Reb02, S. 38f]). Primärenergiequellen stellen die dazu notwendige Energie bereit (vgl. [Reb02, S. 36 und Tabelle 1.4]).

Fossile Primärenergiequellen, wie Kohle, Öl und Erdgas, sind in ihren Vorkommen endlich (vgl. [Reb02, S. 68]). Durch die Freisetzung von CO<sub>2</sub> und anderen Stoffen bei der Umwandlung dieser Energiequellen gelten diese inzwischen zudem als umweltschädlich (vgl. [Reb02, S. 56]). Auch die Nutzung atomarer Energie als Primärenergiequelle stellt sich als problematisch dar. Vor dem Hintergrund des mehrfachen atomaren Super-GAU, wie die Kernschmelzen in den Atomkraftwerken Tschernobyl im April 1986 und Fukushima im März 2011 suchen viele Staaten inzwischen verstärkt nach Alternativen (vgl. [Sch18, S. 157ff]).

Deutschland hat mit der *Energiewende* einen Weg von fossilen und atomaren Energiequellen hin zu erneuerbaren Primärenergiequellen eingeschlagen und gilt damit vielen anderen Ländern als Vorbild. Im Rahmen der Energiewende wurden unter anderem Ausbauziele für den Anteil an erneuerbaren Energiequellen, Klimaschutzziele zur Reduzierung des Ausstoßes an Treibhausgasen wie CO<sub>2</sub> sowie Ziele zur Erhöhung der Energieeffizienz definiert (vgl. [Bun10], [Bun13], [Bun18, S. 7]). Mit der *Erhöhung der Energieeffizienz* soll der Quotient aus gewonnener Leistung oder Arbeit zum eingebrachten Energieeinsatz verbessert werden (vgl. [Eur12, S. 19]).

Das Themengebiet der Optimierung der Energieeffizienz auf Verbraucherseite stellt den Schwerpunkt dieser Arbeit dar. Dies wird dem *Energiemanagement* zugerechnet. Dabei schließen die hier fokussierten Umgebungen alle örtlichen immobilen Umgebungen ein, in denen Energie eingesetzt wird, wie dies beispielsweise in Privaträumen, Büros, Laboren, Werkstätten oder auch gemeinschaftlich genutzten Einrichtungen geschieht. Zusammenfassend werden hier alle Anlagen, Maschinen oder Geräte, die in diesem Sinne Energie

umsetzen, fortan als *Geräte* bezeichnet. Laut [Klo+06] ist unter dem Begriff Energiemanagement „die Kontrolle und Koordination aller Aktivitäten [zu verstehen], die zur Deckung eines bestimmten Energiebedarfs beitragen“ (siehe [Klo+06, S. 117]). Unter dem Begriff Energiemanagement sind also Tätigkeiten zu verstehen, die sich mit dem Energieeinsatz von Geräten beschäftigen und das Ziel verfolgen, Geräte in Abstimmung mit anderen Geräten und Parametern der Umgebung optimal zu betreiben. Diese Tätigkeiten können sich in verschiedenen Kontexten, wie in Privathaushalten oder in der Industrie, deutlich unterscheiden. Im privaten Kontext beziehen sich die Tätigkeiten unter anderem auf den Austausch veralteter Geräte durch energiesparendere Modelle (vgl. [PD11, S. 381 und 383]). Im industriellen Kontext werden darunter insbesondere Geschäftsprozesse, also definierte Abläufe innerhalb eines Unternehmens verstanden (vgl. [Kah+12, S. 16]). Weiter bekräftigt [Klo+06], dass es das Ziel dieser Aktivitäten sei, „einen möglichst effizienten Energieeinsatz zu gewährleisten“ (siehe [Klo+06, S. 117]). Bezüglich des Geräteeinsatzes ist damit die Verringerung der Menge der eingesetzten Energie zu verstehen. Diese Verringerung wird auch mit dem physikalisch nicht korrekten Begriff *Energiesparen* bezeichnet. Aus Sicht der Energieeinspeisung kann der Energieeinsatz auch durch dessen zeitliche Verschiebung effizienter gestaltet werden. Steht beispielsweise durch einen hohen Ertrag einer lokalen Photovoltaikanlage ausreichend Energie zur Verfügung, so könnten Geräte bestimmte Aufgaben zur verbesserten Ausnutzung der zur Verfügung stehenden Energie zeitlich vorziehen (vgl. [Klo+06, S. 122]). Durch die Fähigkeit, Aufgaben oder Dienste von Geräten zeitlich zu verschieben, kann damit ggfs. ein Beitrag zur Vermeidung erforderlicher Ausbaumaßnahmen an Energieeinspeise- oder Übertragungskapazitäten geleistet werden (vgl. [Klo+06, S. 131]). Zusammenfassend lässt sich feststellen, dass das Energiemanagement verschiedene Aspekte der Optimierung zum Ziel hat und damit aufzeigt, dass erst durch die Kombination unterschiedlicher Tätigkeiten das Optimum gehoben werden kann.

Diese Arbeit konzentriert sich auf den Bereich der Informations- und Kommunikationstechnik (IKT) zur Unterstützung des Energiemanagements. Entsprechende technische Systeme werden als *Energiemanagementsysteme* bezeichnet. Diese Systeme unterstützen Besitzerinnen und Besitzer bzw. Anwenderinnen und Anwender dabei, den Energieeinsatz ihrer Geräte zu optimieren, um damit z. B. Kosten zu sparen (vgl. auch [PD11, S. 381] und [BZ15, S. 320]). Dafür können unterschiedliche *Maßnahmen zum Energiemanagement* gewählt werden, die sich an den oben genannten Tätigkeiten orientieren (siehe ausführlich dazu Abschnitt 2.1). Durch die Maßnahmen können beispielsweise vernetzte Geräte aufeinander abgestimmt werden, um eine Glättung des Gesamtenergiebedarfs vorzunehmen. Als weitere Maßnahme kann der Anwenderin oder dem Anwender eine Visualisierung des Energieeinsatzes der Geräte bereitgestellt werden, damit diese daraus Rückschlüsse auf effizientere Nutzungen ziehen können (siehe ausführlicher dazu Abschnitt 2.2).

Wie beschrieben, können Energiemanagementsysteme in vielen Umgebungen zum Einsatz gelangen. Dabei unterscheiden sich die Geräteausstattungen auch ähnlicher Umgebungen oft deutlich voneinander. So besitzen beispielsweise zwei Küchen verschiedener Haushalte in der Regel unterschiedliche Geräte, obwohl sie der gleichen Funktion dienen. Durch die hohe Variabilität der Umgebungen, sowie die verschiedenen benannten Maßnahmen zum Energiemanagement, ergeben sich viele Freiheitsgrade, die bei der Realisierung von Energiemanagementsystemen zu berücksichtigen sind.

Aus dem gleichen Grund unterscheiden sich auch bereits existierende Lösungen deutlich voneinander. In der Literatur findet sich eine Vielzahl an Konzepten und Algorithmen zur Realisierung von Energiemanagementsystemen (siehe Abschnitt 2.2). Auch am Markt wird eine Vielzahl unterschiedlicher Lösungen angeboten. Bei vielen dieser Konzepte und Systeme ist festzustellen, dass sie nur auf einzelne Maßnahmen zum Energiemanagement abzielen oder nur auf bestimmte Geräteausstattungen eingehen und damit Abhängigkeiten in ihrer Einsatzfähigkeit unterliegen. Häufig stellen diese Systeme zudem Insellösungen dar, die nur für bestimmte, oft hoch spezialisierte, Umgebungen und Geräteausstattungen geeignete Lösungen darstellen. Durch ihre geschlossene oder proprietäre Architektur sind diese Systeme im Regelfall nicht durch Dritte erweiterbar und damit auch nicht auf bestimmte Umgebungen anpassbar. Da die Systeme durch diese Architektur in der Regel auch nicht miteinander kompatibel sind, ist auch deren Kombination im Allgemeinen technisch bisher nicht möglich. Der gleichzeitige parallele Betrieb mehrerer Systeme und der dazu notwendigen Infrastrukturen, würde zudem die Investitions- und Betriebskosten des Energiemanagements erheblich steigern und wäre somit weder technisch noch wirtschaftlich sinnvoll. (siehe ausführlicher zu dieser Problematik Abschnitt 3.2)

Folglich wird eine Lösung benötigt, die unterschiedliche Maßnahmen zum Energiemanagement unter Nutzung vorhandener Geräte und Infrastrukturen **flexibel** in verschiedenen heterogenen Umgebungen ermöglicht. Ziel dieser Arbeit ist es, eine solche Lösung in Form eines **universellen** und **flexiblen Systemkonzepts** zu entwickeln, das als Basis zukünftiger Energiemanagementsysteme dienen kann. Dazu wird ein Schichtenmodell zugrunde gelegt, das die oben beschriebenen Abhängigkeiten voneinander entkoppelt. Das Systemkonzept besteht aus verschiedenen Komponenten und Verfahren, die folgendermaßen begründet werden:

1. Viele bisherige Energiemanagementsysteme sind als zentralisierte Systeme konzipiert und befinden sich demzufolge nicht in unmittelbarer Nähe der Geräte (siehe Abschnitt 2.2). Hierbei werden Daten teilweise über weite Distanzen versendet, teilweise sogar über das Internet. Dadurch können unter Umständen Anforderungen an den Datenschutz und die Datensicherheit nur bedingt gewährleistet werden. Mit dem

- Vorschlag eines neuen Kommunikationsansatzes sollen die Daten deshalb möglichst auf lokalen Instanzen verarbeitet werden. Diese Instanzen teilen Daten mit übergeordneten Instanzen nur noch dann, wenn dies für eine von der Nutzerin bzw. dem Nutzer gewünschte Funktion erforderlich ist. Dadurch bildet sich eine sogenannte **hierarchische Kommunikationsstruktur** aus. (siehe ausführlich dazu Abschnitt 4.3)
2. Die genannten Maßnahmen zum Energiemanagement sollen in Form von separaten Software-Anwendungen realisiert werden, so dass diese variabel in verschiedenen Umgebungen eingesetzt werden können. Die zuvor benannte lokale Datenverarbeitung impliziert, dass diese Anwendungen ebenfalls in den lokalen Instanzen ausgeführt werden müssen. Dazu wird eine **Softwarekonzeption nach einer verteilten Softwarearchitektur** vorgeschlagen, die es erlaubt eine Anwendung in Software-Elemente aufzuteilen. Während die einzelnen Software-Elemente die sensiblen Daten ausschließlich lokal verarbeiten, agiert die Gesamtanwendung als Zusammenspiel der Software-Elemente weiterhin global. (siehe ausführlich dazu Abschnitt 4.3)
  3. Die Einbindung einer möglichst großen Anzahl an Geräten stellt die größte Herausforderung für die Etablierung eines feingranularen Energiemanagements dar. Auch hier wird ein modularer Ansatz verfolgt. In dieser Arbeit wird definiert, welche Daten und Funktionen zum **notwendigen Funktionsumfang** eines Gerätes gehören. In der Praxis haben sich *intelligente Geräte*, die Daten über ihren Betriebsstatus und ihren Energieeinsatz an einer Daten-Schnittstelle bereitstellen, noch nicht flächendeckend etabliert. Als Übergangslösung werden Brückentechnologien vorgeschlagen, die zwei Strategien verfolgen: Einerseits soll die Einbindung nicht vernetzter Geräte durch die Ermittlung von Zustandsdaten aus den Energieverbrauchsdaten ermöglicht werden. Andererseits wird für bereits vernetzte Geräte, die keine Energieverbrauchsdaten zur Verfügung stellen, die Anwendung einer Energieverbrauchsschätzung vorgeschlagen, welche auf bekannte, zuvor ermittelte, Daten zurückgreift. (siehe ausführlich dazu Abschnitt 4.1 und 4.2)
  4. Aus den genannten Maßnahmen zum Energiemanagement mit und ohne Nutzerunterstützung werden **Zielfunktionalitäten** abgeleitet, die als Teil des Systemkonzepts umgesetzt werden sollen. Dabei wird dargelegt, wie die Umsetzung in der verteilten Struktur erfolgen kann. Die Funktionalitäten bestehen analog zu den oben genannten Maßnahmen sowohl aus Ansätzen zum automatisierten Energiemanagement, aber auch aus Ansätzen, die Nutzerinnen und Nutzer beim Energiemanagement unterstützen sollen. (siehe ausführlicher dazu Abschnitt 4.4 bis 4.6)

Zunehmend werden bislang nicht vernetzte Gegenstände und Geräte miteinander verbunden und damit neue Anwendungen ermöglicht. Als Beispiele sei an dieser Stelle auf das *Smart Home* und die *Industrie 4.0* für die intelligente Vernetzung von Privathaushalten bzw. in der Industrie verwiesen. Das Ziel der *intelligenten Vernetzung* ist es dabei, mittels neuartiger Software-Anwendungen neue Anwendungsgebiete zu erschließen. So könnte ein mit entsprechenden Softwarefunktionen ausgerüsteter Kühlschrank zukünftig in der Lage sein, selbstständig neue Waren zu bestellen. Das *Internet der Dinge* (engl. Internet of Things, IoT) verfolgt sogar einen noch allgemeineren Ansatz. Hier sollen bisher nicht vernetzte Gegenstände an das Internet angeschlossen werden und dadurch anderen Teilnehmern ihre Dienste zur Verfügung stellen. (vgl. [MBR15, S. 72-74], siehe ausführlicher dazu Abschnitt 2.3) Angesichts dieser kommenden Veränderungen bietet es sich an, zukünftig Energiemanagementsysteme als integrale Bestandteile der IoT-Infrastrukturen zu realisieren. Diese Vision wird in dieser Arbeit deshalb bewusst ins Auge gefasst. Das Systemkonzept versteht sich dabei als übergreifender Lösungsansatz für verschiedene Anwendungsbereiche. Dabei wird der Schwerpunkt jedoch auf Energiemanagementsysteme gelegt und Aspekte einer umfassenden IoT-Infrastruktur werden nur am Rande betrachtet.

In dieser Arbeit wird ausschließlich auf die Energieform der elektrischen Energie eingegangen. Viele der Theorien und Erkenntnisse dieser Arbeit können jedoch direkt auf andere Energieformen übertragen werden.

Um das Ziel dieser Arbeit zu erreichen, wird das nachfolgend beschriebene Vorgehen gewählt, welches auch den Aufbau dieser Arbeit widerspiegelt:

- In **Kapitel 2** werden der Stand der Technik und die Rahmenbedingungen vorgestellt: Dabei werden das Energiemanagement, seine Stakeholder und mögliche Maßnahmen zum Energiemanagement beschrieben. Anschließend werden vorhandene Lösungen zu Energiemanagementsystemen bzw. -konzepten anhand von Literaturzitaten vorgestellt. Dabei wird die Architektur der Systeme besonders betrachtet. Weiterhin werden Grundlagen und aktuelle Entwicklungen zu Kommunikationsstrukturen insbesondere in Hinblick auf das Internet der Dinge zusammengefasst.
- In **Kapitel 3** wird aus dem Stand der Technik eine Vision für zukünftige Energiemanagementsysteme abgeleitet. Weiterhin werden Defizite existierender Lösungsansätze zusammengetragen. Im gleichen Kapitel wird ein Zielsystem zur Realisierung der Vision beschrieben. Zur Überwindung der Defizite setzt dieses System auf dem genannten Schichtenmodell auf. Aus der Beschreibung des Zielsystems werden weiterhin Anforderungen zur Überprüfung der Zielerreichung abgeleitet und Zielfunktionalitäten aufgestellt.

- Der Kern dieser Arbeit ist die Entwicklung eines offenen und flexiblen Systemkonzepts. Dieses wird auf Basis des Schichtenmodells in **Kapitel 4** beschrieben. Dabei werden die oben benannten Komponenten und Verfahren für die jeweiligen Schichten vorgestellt.
- In **Kapitel 5** wird die erarbeitete Implementierung erläutert. Sie soll zeigen, dass das Systemkonzept in der Praxis realisierbar ist und damit die Defizite bisheriger Energiemanagementsysteme überwunden werden können. Die Implementierung wurde in Form einer Softwareplattform umgesetzt, welche Software-Anwendungen nach einer verteilten Softwarekonzeption ausführen kann. Zusätzlich werden u. a. die Schnittstellen zum Datenaustausch und das Protokoll zur Kommunikation in der hierarchischen Kommunikationsstruktur beschrieben. Für alle genannten Zielfunktionalitäten des Energiemanagements werden die jeweiligen Implementierungen für die Softwareplattform beschrieben. Auch verschiedene Techniken zur Einbindung von bestehenden Geräten auf Basis der Signalverarbeitung und -erkennung werden vorgestellt.
- In **Kapitel 6** werden der Einsatz und die Evaluation der Implementierungen in realen Umgebungen beschrieben. Mittels verschiedener Demonstratoren werden die beschriebenen Komponenten des Systemkonzepts einzeln und in Kombination demonstriert und erprobt. Die Umgebungen wurden mit einer industriellen Werkzeugmaschinen-Halle, der sogenannten Forschungsfabrik, mehreren Instituten mit Büroräumen, Laboren und Teeküchen, aber auch mit Privathaushalten bewusst unterschiedlich gewählt. Die im Rahmen dieser Arbeit entstandenen Konzepte werden anschließend einzeln evaluiert. Die Leistung der Kommunikationsstruktur wird anhand von qualitativen Bewertungen, aber auch durch Belastungs- und Latenzmessungen analysiert. Die Techniken zur Geräteeinbindung werden quantitativ überprüft.
- Abschließend werden die Erkenntnisse der Arbeit in **Kapitel 7** zusammengefasst und dieses mit einem Ausblick abgeschlossen.

## 2 Stand der Technik

In diesem Kapitel wird der Stand der Technik im Bereich der Energiemanagementsysteme und der Aufbau solcher Systeme dargelegt. Weiterhin wird auf verwandte Systeme und Infrastrukturen eingegangen.

Damit das Themengebiet des Energiemanagements in seiner Breite verständlich gemacht werden kann, wird in Abschnitt 2.1 vorgestellt, welche Beweggründe zum Energiemanagement von verschiedenen Stakeholdern verfolgt werden. Diese Beweggründe sind unterschiedlich und teilweise sogar gegensätzlich. Maßnahmen, die für das Energiemanagement genutzt werden, sind folglich sehr vielfältig. Sie reichen von kurzfristigen Steuerungseingriffen mittels automatisierter Systeme bis zum dauerhaften Austausch von ineffizienten Geräten durch informierte Nutzerinnen und Nutzer.

Durch die Vielfältigkeit der Maßnahmen haben sich verschiedene Arten an Energiemanagementsystemen entwickelt. In Abschnitt 2.2 wird zunächst zwischen zwei grundsätzlichen Systemansätzen unterschieden. Die Systemansätze werden anschließend anhand von Systembeschreibungen in verschiedenen Anwendungsbereichen erläutert. Dabei wird anhand aktueller Literaturstellen auch der Stand der Forschung zu den jeweiligen Ansätzen dargelegt.

Basis für das Energiemanagement sind die Kenntnis des Energieverbrauchs, aber auch weitere Informationen zu Eigenschaften und Zuständen von Geräten und Maschinen. Entsprechend ist eine Kommunikationsanbindung möglichst vieler ausgewählter Geräte in einer betreffenden Umgebung sinnvoll. Deshalb werden in Abschnitt 2.3 Infrastrukturen zur Anbindung von Geräten und Sensoren mit Fokus auf das Internet der Dinge betrachtet. Dabei werden insbesondere die zugrundeliegenden Kommunikationsstrukturen betrachtet.

### 2.1 Energiemanagement

Bereits seit den 1960er Jahren gibt es erste praktische Realisierungen zur Überwachung und Steuerung des Energieversorgungsnetzes und seiner Energieerzeuger. Mit der zunehmenden Einführung von Fernwirkanlagen in den 1980er-Jahren werden seitdem auch verschiedenste

Techniken zur Beeinflussung der Energieverbraucher aus Richtung des Energieversorgungsnetzes entwickelt. Ein Beispiel ist hier die Tonfrequenz-Rundsteuerung zur Steuerung von Nachtstromspeicherheizungen oder die Fern-Ein- bzw. Ausschaltung der Straßenbeleuchtung. (vgl. [Klo+06, S. 120]) Diese Techniken werden dem Bereich des *Demand Side Managements* (engl. für Management auf der Verbraucherseite, DSM) zugerechnet. Die Quelle [PD11] versteht unter DSM *alle* Aktionen auf Seiten der Energieverbraucher, vom Austausch von ineffizienten Geräten bis zur Installation von vollständig automatisierten Steuerungssystemen. Diese im Vergleich zu anderen Quellen sehr allgemeine Definition soll zur Erläuterung des Energiemanagements verwendet werden, um alle Möglichkeiten zum Eingriff in den Energieverbrauch darstellen zu können.

Diese Arbeit betrachtet das Management des Verbrauchs und der Erzeugung elektrischer Energie auf Verbraucherseite (siehe Kapitel 1). Deshalb werden in Unterabschnitt 2.1.1 zunächst die Grundprinzipien der elektrischen Energieversorgung beschrieben. Aus diesen Grundprinzipien werden in Unterabschnitt 2.1.2 die Beweggründe zum Energiemanagement aus Sicht der Stakeholder des Energieversorgungsnetzes abgeleitet. In Unterabschnitt 2.1.3 werden hierzu noch die Sichtweisen der Endverbraucher hinzugefügt. Dabei wird auf die oben genannte allgemeine Definition des Energiemanagements zurückgegriffen, um alle Möglichkeiten zu erfassen. Darauf aufbauend wird in Unterabschnitt 2.1.4 dargelegt, welche praktischen Maßnahmen beim Energiemanagement ergriffen werden können, um den Energieeinsatz zu optimieren.

### 2.1.1 Grundprinzipien der elektrischen Energieversorgung

Die elektrische Energie ist die einzige mediumlose Energieform, die in großen Mengen auch über weite Strecken transportiert werden kann. Die entstandenen elektrischen Energieversorgungsnetze decken dabei Länder, aber auch ganze Kontinente ab. Sie wird als nutzbare Energieform aus der Umwandlung aus anderen Energieformen gewonnen, sogenannten Primärenergiequellen z. B. aus Kohle, Öl, Sonneneinstrahlung oder Windenergie.

Die Umwandlung und der Einsatz von Energie werden physikalisch nicht korrekt auch als „*Energieerzeugung*“ und „*Energieverbrauch*“ bezeichnet. Der Einfachheit halber werden diese Begriffe auch in dieser Arbeit für die Energieumwandlung in bzw. aus nutzbaren Energieformen verwendet.

Elektrische Energie kann im Gegensatz zu anderen Energieformen nur mit hohem technischen Aufwand und in vergleichsweise geringen Mengen gespeichert werden (vgl. [Lin10]). Die Energieerzeugung und der Energieverbrauch müssen deshalb netzweit jederzeit möglichst



genau übereinstimmen, da es sonst zur Netzinstabilität oder sogar zum vollständigen Black-out kommen kann. Marktinstrumente wie die *Leipziger Strombörse EEX* (engl. European Energy Exchange)<sup>1</sup> sorgen heute u. a. dafür, dass Großkunden und Energievertrieb möglichst genau so viel Energie einkaufen, wie sie bzw. ihre Kunden benötigen. Dazu bieten die Kraftwerksbetreiber im Vorhinein ihre potentielle Erzeugungsleistung zum Kauf an, sodass Angebot und Nachfrage miteinander abgeglichen werden können. (vgl. [GOZ08] sowie EnWG § 1)

Zur Korrektur der verbleibenden Differenzen und für den Fall, dass Störungen auftreten, sorgt eine mehrstufige Netzregelung für die Netzstabilität. Vorher bestimmte Kraftwerke stellen dabei sogenannte Regelleistung zur Verfügung, die bei Bedarf schnell ins Netz eingespeist werden kann. (vgl. [Ver07])

Zur Erhöhung der Wettbewerbsfähigkeit wurden in Deutschland nach EnWG § 8 (Eigentumsrechtliche Entflechtung) die Aufgaben der elektrischen Energieversorgung auf verschiedene unabhängige Unternehmen aufgeteilt. So wird insbesondere zwischen Energievertriebsunternehmen und Netzbetreibern unterschieden.

*Energievertriebsunternehmen* handeln nach wirtschaftlichen Gesichtspunkten mit Energiemengen. So kaufen sie Energie beispielsweise an der Leipziger Strombörse EEX und bieten ihren Kunden pauschalierte oder individuelle Verträge zum Bezug elektrischer Energie an. Zu den Kunden zählen Privatkundinnen und -kunden, aber auch Sondervertragskunden aus der Industrie und öffentlichen Einrichtungen. Sie werden nachfolgend zusammengefasst als *Endverbraucher* bezeichnet.

*Netzbetreiber* betreiben ein Netz und sorgen dafür, dass vereinbarte Leistungen jederzeit übertragen werden können. Sie sind in ihrer Arbeit stark durch Gesetze reglementiert (siehe EnWG §§ 11ff) und unterliegen der Aufsicht der Bundesnetzagentur. Die Netze werden dabei grundsätzlich nach Ebenen in Übertragungsnetze und Verteilnetze unterschieden. Entsprechend gibt es Übertragungsnetzbetreiber und Verteilnetzbetreiber. Die Netzbetreiber selbst werden durch die sogenannten *Netznutzungsentgelte* finanziert, die u. a. in dem Strompreis der Endverbraucher beinhaltet sind (siehe EnWG § 9 Absatz 3).

Der Betrieb der Netze schließt auch die Systemverantwortung mit ein. Dabei sind die Übertragungsnetzbetreiber für die Netzstabilität verantwortlich (siehe EnWG § 13). Die Bereitstellung der Regelleistung wird von den Übertragungsnetzbetreibern ausgeschrieben bzw. eingekauft.

Die Netzbetreiber sind darüber hinaus dazu berechtigt, u. a. Betreiber von Erzeugungsanlagen (z. B. Windkraftanlagen) aufzufordern ihre Einspeisung zu reduzieren, um die Netzstabilität

---

<sup>1</sup>Siehe <https://www.eex.com/de/>

zu gewährleisten (siehe EnWG § 13 und EEG 2017 § 14). Auch bei drohenden lokalen Überlastungen von Betriebsmitteln im Übertragungs- oder Versorgungsnetz, sogenannten Netzengpässen, dürfen die Netzbetreiber von diesem Recht Gebrauch machen. Die Betreiber von EEG-Anlagen erhalten dafür eine Vergütung der entgangenen Erlösmöglichkeiten nach EnWG § 13a.

Die Energiewende (siehe Kapitel 1) führt zu großen Veränderungen der Energieversorgungsnetze in Deutschland. So sind bereits heute in großem Umfang Erzeuger erneuerbarer Energie, wie beispielsweise Photovoltaikanlagen, Windkraftanlagen und Biogaskraftwerke an das Energieversorgungsnetz angeschlossen. Die Energieeinspeisung wird den Anlagenbetreibern in Deutschland durch die sogenannte Einspeisevergütung oder ab einer bestimmten Einspeiseleistung durch eine Marktprämie entlohnt (siehe EEG 2017 § 19). Dies wird durch das Erneuerbare-Energien-Gesetz (EEG 2017) geregelt. Diese Vergütung bzw. Prämie wird per Umlage, der sogenannten EEG-Umlage, über den Strompreis auf die Kunden umgelegt (siehe EEG 2017 § 60).

Endverbraucher, die sowohl elektrische Energie beziehen, aber auch erzeugen, werden auch als *Prosumer* bezeichnet (siehe auch [Rod+14, S. 6143] und [Lut+16, S. 2]). Das Kunstwort setzt sich aus den englischen Worten „Consumer“ für Konsument und „Producer“ für Produzent zusammen. Nicht direkt lokal verbrauchte Energie wird dabei zurück in das Energieversorgungsnetz eingespeist. In Deutschland wurden durch politische Rahmenbedingungen weitere Anreize geschaffen, so dass z. B. der Eigenverbrauch von selbst erzeugter erneuerbarer Photovoltaik-Energie eine höhere Vergütung als die Einspeisung erbringt (vgl. [Lut+16, S. 9], EEG 2017 § 61b). Dies soll Prosumer dazu motivieren, durch entsprechende Eingriffe ihren Energieverbrauch mit ihrer Energieerzeugung bereits auf lokaler Ebene abzugleichen.

Erzeuger erneuerbarer Energie sind vielfach nicht wie Großkraftwerke möglichst an den Zentren erhöhter Energieabnahme gelegen, sondern in der Regel ländlich und somit dezentral angesiedelt. Des Weiteren ist ihre Anzahl erheblich höher als die Zahl klassischer Kraftwerke, was bedeutet, dass ihre Anbindung und Ansteuerung erheblich aufwändiger ist. Die Vielzahl dieser Anlagen gewinnen ihre elektrische Energie aus Primärenergien, deren Menge teilweise stark über die Zeit variiert, zum Beispiel aus der Sonneneinstrahlung oder dem Wind. Entsprechend variiert auch im gleichen Maß deren Energieerzeugung. Da der Prozess der Energieumwandlung aus Solarenergie und Wind nicht zeitlich verzögert werden kann und die Energien kaum zwischengespeichert werden können, ist eine Steuerung dieser Erzeuger nach den Verbrauchsmengen, wie bei den konventionellen Großkraftwerken üblich, nur sehr bedingt möglich. Bei diesen Anlagen ist in der Regel lediglich ein Aussetzen bzw. Herabsetzen der Energieumwandlung möglich, wodurch jedoch die mögliche Einspeisemenge ungenutzt bleibt (vgl. [Amp+16, S. 4]). Dieser Prozess kommt bei Netzengpässen zum Einsatz und wird als

Abregelung bezeichnet. Wie bereits beschrieben, wird den Betreibern in Deutschland dieser Verlust nach ENWG § 13a durch eine Umlage vergütet. Auch der Umwandlungsprozess in Biomasse-Kraftwerken kann nur bedingt beeinflusst werden. Prinzipiell kann das entstehende Rohgas auch mit einer entsprechenden Gasaufbereitung in zusätzliche Gasspeicher verbracht oder durch direkte Einspeisung in das Gasnetz eingespeist werden.

Die EU hat sich das auf EU-Ebene verbindliche Ziel gesetzt, bis zum Jahr 2030 mindestens einen Anteil von 32 % ihres Energieverbrauchs aus erneuerbaren Energien zu erzeugen (vgl. [Eur14, S. 5], [Eur18b, S. 6]). Die Region EU 28 erzeugte im Jahr 2016 bereits 29,6 % ihres Bruttostromverbrauchs aus erneuerbaren Energien (vgl. [Bun18, S. 41]). Deutschland möchte bis zum Jahr 2030 sogar 65 % des Bruttostromverbrauchs aus erneuerbaren Energien erzeugen (vgl. [Bun18, S. 7]). In Deutschland wurden im Jahr 2017 bereits 36 % des Bruttostromverbrauchs aus erneuerbaren Energien erzeugt (vgl. [Bun18, S. 9]). Aus diesen Zahlen wird deutlich, dass der Anteil der bisher nicht oder kaum steuerbaren Energieerzeugung deutlich zugenommen hat und weiter zunehmen wird. Zur Gewährleistung der Netzstabilität ist es erstrebenswert, Steuerungsfunktionen für dezentrale Erzeugungsanlagen, aber auch für die Energieverbraucher zu schaffen. Insgesamt erscheint eine intelligente Nutzung immer dringender geboten. In den nachfolgenden Unterabschnitten soll deshalb dargelegt werden, welche Beweggründe und Möglichkeiten es zum Management von Energie gibt.

### 2.1.2 Beweggründe für das Energiemanagement

Anhand der allgemeinen Definition des Energiemanagements in der Einleitung zu Abschnitt 2.1 aus [PD11] sollen die Beweggründe der verschiedenen Stakeholder zum Energiemanagement auf Verbraucherseite zusammengetragen werden. Abbildung 2.1 zeigt eine Einordnung verschiedener Varianten des Demand-Side-Managements im Vergleich. In der Abbildung sind die Varianten über die Zeitspannen des (Management-)Eingriffs sowie über die Art der folgenden Energieverbrauchsreduzierung aufgetragen. Dauerhafte und kurzfristige Reduktionen führen entsprechend zu dauerhaften bzw. kurzfristigen Vorteilen. Beim optimierten Ablauf werden durch viele einzelne Eingriffe ebenfalls längerfristige Vorteile erreicht. Die Varianten werden im weiteren Verlauf separat erläutert. (vgl. [PD11, S. 381], [KCM12, S. 4])

Anstatt neue Kraftwerke zu errichten oder vorhandene zu ersetzen, wäre es gesamtwirtschaftlich deutlich sinnvoller, die *Energieeffizienz* des Energieeinsatzes **bei Industrie- und Endverbrauchern** zu erhöhen. Mit einer Erhöhung der Energieeffizienz wäre bei gleicher Leistung der Geräte (z. B. Kälte, Licht, Wärme, ...) weniger Energieeinsatz erforderlich. Dadurch könnte der Gesamtenergieverbrauch reduziert werden und auf Erzeugungskapazitäten verzichtet werden. Entsprechend hat sich die Europäische Union zum Ziel gesetzt, bis

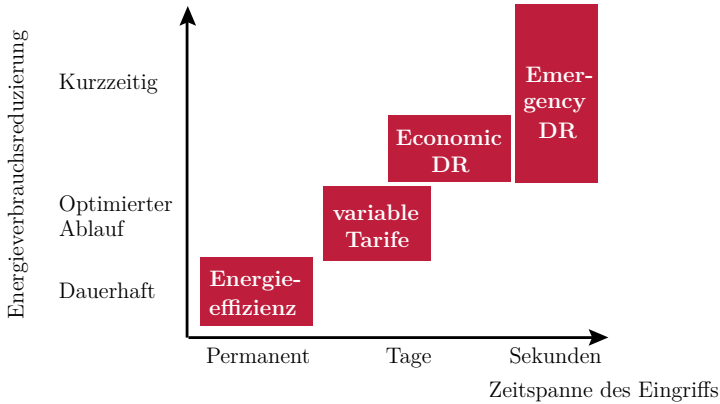


Abbildung 2.1: Varianten des Demand-Side-Managements aufgetragen über die Zeitspanne des Eingriffs und über die Art der folgenden Energieverbrauchsreduzierung (nach [PD11, Bild 1]). Economic und Emergency DR stehen für zwei Varianten des Lastmanagements (engl. Demand Response). Die Art der Energieverbrauchsreduzierung beschreibt dabei, wie lange die Vorteile aus den unterschiedlichen Varianten gezogen werden können. Beim optimierten Ablauf werden die Vorteile aus vielen einzelnen Eingriffen gezogen.

zum Jahr 2030 mindestens 32,5 % ihres Primärenergieverbrauchs einzusparen (vgl. [Eur14, S. 6], [Eur18a, S. 7]). Auch die Bundesregierung hat mit ihrem Energiekonzept (vgl. [Bun13, S. 6f]) das Ziel gesetzt, den Energieeinsatz zu reduzieren. Maßnahmen zur Erhöhung der Energieeffizienz sind unter anderem der Austausch von ineffizienten Geräten oder auch die Dämmung von Wohngebäuden. Auf staatlicher Ebene gibt es bereits verschiedene Förderprogramme, die entsprechende Maßnahmen zusätzlich attraktiv machen sollen. Auch die Kennzeichnungspflicht von Haushaltsgeräten in der EU, beispielsweise mit dem Label A bis G, ist eine Maßnahme zur Steuerung der Energieeffizienz (siehe EU 2017/1369 bzw. EnVKG). Die Quelle [PD11] zählt die Erhöhung der Energieeffizienz zu den Varianten des Demand Side Managements. Wie in Abbildung 2.1 dargestellt, werden unter der Erhöhung der Energieeffizienz Maßnahmen eingeordnet, die einen permanenten zeitlichen Charakter aufweisen und dabei den Energieverbrauch dauerhaft optimieren.

Aus wirtschaftlicher Perspektive gibt es darüber hinaus verschiedene Stakeholder, die ein Interesse an der Beeinflussung des Gesamt- bzw. Einzel-Energieverbrauchs bei Endverbrauchern (bzw. ggfs. auch Eigentümern) haben (siehe Unterabschnitt 2.1.1): Energievertriebsunternehmen, Netzbetreiber, Energieerzeuger sowie Prosumer. Deren Beweggründe sollen im Folgenden dargelegt werden. (Definitionen siehe auch EnWG § 3)

**Energievertriebsunternehmen** handeln, wie bereits erläutert, nach wirtschaftlichen Aspekten. Sie müssen sich dem Wettbewerb in Privathaushalten, Unternehmen und öffentlichen Einrichtungen stellen. Durch die Beeinflussung des Energieverbrauchs ihrer Kunden könnten sie ggfs. in laststarken Verbrauchssituationen auf teure Energiezukäufe verzichten oder diese zumindest reduzieren. Bereits heute werden Sondervertragskunden (insbesondere aus der Industrie) bei derartigen Beeinflussungen Preisvorteile angeboten. Solche Preisvorteile könnten zukünftig auch Privatkunden eingeräumt werden. Die Eingriffsmöglichkeiten der Versorger in den Energieverbrauch ihrer Kunden lassen sich in zwei Kategorien einteilen:

- *Variable Tarife*: Beispielsweise durch (Tages- oder Jahres-) Zeit-abhängige Tarifgestaltungen (engl. time of use pricing, TOU) werden bestimmte Verbrauchszeiträume (z. B.: 17:00 bis 19:00 Uhr) mit höheren Arbeitspreisen belegt. Die Kunden sollen dadurch motiviert werden, die Nutzung ihre Geräte in lastschwächere Tageszeiten zu verschieben. Diese Eingriffs-Variante kann bis zur Echtzeit-Tarifgestaltung (engl. Real-Time Pricing, RTP) erweitert werden. Die Literatur benennt weiterhin noch die Tarifgestaltung bei kritischen Lastspitzen (engl. Critical Peak Pricing, CPP), einer abgeschwächten Variante bei der in Situationen mit hoher Last veränderte Tarife berechnet werden. Die variablen Tarife sind nach Abbildung 2.1 durch eine Eingriffszeit im Tagesverlauf charakterisiert und können durch viele einzelne Eingriffe zu einer Energieverbrauchsreduktion in Form eines optimierten Ablaufs führen. Eine dauerhafte Energieersparnis ist durch variable Tarife nicht zu erwarten. (vgl. [PD11, S. 382], [AE07, S. 2], [Sha+18, S. 24501]). In Deutschland müssen Energievertriebsunternehmen nach EnWG 2015 § 40 (5) ihren Kunden einen solchen Tarif anbieten, sofern dies wirtschaftlich zumutbar und technisch machbar ist.
- *Direkteingriff*: Mittels der Technik des sogenannten *Demand Response* (engl. für Lastmanagement, DR) gewähren Eigentümer (zur Zeit insbesondere Industriekunden) dem Energievertriebsunternehmen direkten Zugriff auf einige ihrer Prozesse, um daraus gegenseitige wirtschaftliche Vorteile zu erzielen. Bei dem Zugriff kann es sich sowohl um gezielte Abschalt- aber auch Zuschaltanweisungen handeln. Diese Variante des Demand Response wird auch als *Market Demand Response* oder *Economic Demand Response* bezeichnet. Sie wird den Sondervertragskunden durch die Energievertriebsunternehmen vergütet. Wie in Abbildung 2.1 dargestellt, besitzt diese Variante kurzfristigere Eingriffszeiten als die variablen Tarife. Entsprechend ist aber auch die Reduktion kurzzeitiger. Auch bei dieser Variante ist keine dauerhafte Energieersparnis zu erwarten. (vgl. [PD11, S. 382], [KCM12, S. 3] , [Sha+18, S. 24500ff])

Weiterführend kann eine feinere Untergliederung der Varianten in unterschiedliche Vergütungsmethoden der Quelle [AE07, S. 2] entnommen werden.

Wie in Unterabschnitt 2.1.1 beschrieben, sind **Netzbetreiber** für die Netzstabilität verantwortlich. Im Sinne der Netzregelung oder der Vermeidung von Netzengpässen können dazu auch Lasten als negative (Regel-)Leistung zugeschaltet oder als positive (Regel-)Leistung abgeschaltet werden. Solche automatisierten Steuerungssysteme werden ebenfalls dem Demand Response zugerechnet. Diese Variante wird in der Literatur als *Emergency Demand Response* oder *Physical Demand Response* bezeichnet. Wie in Abbildung 2.1 gekennzeichnet, besitzen Maßnahmen dieser Variante eine Eingriffszeit im Bereich von Minuten bis Sekunden. Die zu erwartende Energieverbrauchsreduktion befindet sich ebenfalls im kurzzeitigen Bereich. (vgl. [PD11, S. 382], [KCM12, S. 3]).

Um in Deutschland an einer automatisierten Zu- bzw. Abschaltung nach EnWG § 13a (siehe Unterabschnitt 2.1.1) teilnehmen zu dürfen – was in etwa dem aus der Literatur beschriebenen Emergency DR entspricht – müssen sich Anlagenbetreiber zunächst dem Prozess der sogenannten Präqualifikation unterziehen. Dieser Prozess ist gesetzlich vorgeschrieben und benennt, welche rechtlichen und technischen Voraussetzungen erfüllt sein müssen, bevor die Anlage an der Netzregelung teilnehmen darf (vgl. [Amp+16], [Amp+18]).

**Energieerzeuger** und damit auch **Prosumer** haben ein grundsätzliches Interesse, eine möglichst hohe Rendite auf ihre eingesetzten Investitionen zu erzielen. Bei steuerbaren Energiequellen oder durch den Einsatz von zusätzlichen Speichern können sie zudem im Rahmen ihres Energiebezugstarifs wirtschaften oder sogar aktiv am Energiehandel teilnehmen. Für Energiespeicher gibt es zusätzliche Förderprogramme (z. B. EEG 2017 § 61k). Da der Eigenverbrauch von selbst erzeugter, erneuerbarer Energie in der Regel rentabler ist als die Einspeisung in das vorgelagerte Netz (siehe Unterabschnitt 2.1.1), ist auch die Motivation bei Prosumern gestiegen, ihren Energieverbrauch beeinflussen zu wollen. Auch Synergieeffekte mit anderen Energiemedien, beispielsweise in Form der Kraft-Wärme-Kopplung oder die Einbindung von Elektrofahrzeugen in das Energiemanagement stehen im Fokus von Prosumern (vgl. [Lut+16] und [Rod+14]).

Neben der Eigenverbrauchsoptimierung eines einzelnen Kunden, wäre zukünftig auch eine selbstständige Abstimmung zwischen weiteren benachbarten Energieverbrauchern und -erzeugern denkbar, um z. B. Kapazitätsengpässe im Verteilnetz zu vermeiden. Hier sind verschiedene Bereiche denkbar: innerhalb eines Mehrparteien-Gebäudes, innerhalb einer Straße, innerhalb eines Quartiers (Stadtteils), innerhalb einer Stadt bzw. eines Landkreises oder innerhalb einer Region. In Deutschland wurde für Mehrparteien-Gebäude durch die sogenannte Mieterstromförderung nach EEG 2017 § 21 (Einspeisevergütung und Mieterstromzuschlag) dazu bereits eine gesetzliche Grundlage zur Vergütung geschaffen (siehe [BMW17]). Auf die gleiche Weise könnte auch eine Abstimmung innerhalb eines Unternehmens mit unterschiedlichen Standorten, das als ein Gesamtkunde auftritt, sinnvoll sein.

Unter dem Begriff des *Smart Grid* (engl. für intelligentes Netz) werden Maßnahmen zusammengefasst, um das Energieversorgungsnetz mittels Kommunikationstechnik intelligenter zu gestalten (vgl. [AE12, S. 13]). Hierbei werden insbesondere die noch nicht vernetzten unteren Netzebenen (Mittel- und Niederspannungsnetze) betrachtet, da dort bisher wenig oder sogar keine Überwachungs- bzw. Steuerungstechnik zum Einsatz kommt. Entsprechend stehen auch Schnittstellen zu Einspeisern und Verbrauchern dieser Netzebenen im Fokus. (vgl. [AE12, S. 135f]) In vielen Schaubildern der betrachteten Literatur ist deshalb das Smart Grid bereits als Schnittstelle zwischen dem Netz und den Systemen des Kunden aufgeführt.

Historisch ist das Energieversorgungsnetz aus großen zentralen Kraftwerken möglichst an Lastschwerpunkten oder an Förderstandorten fossiler Energieträger und einer daran ausgerichteten hierarchischen Netzstruktur aufgebaut (siehe Unterabschnitt 2.1.1 und vgl. auch [PD11, S. 381]). In der Vergangenheit ergaben sich deshalb zunächst netzweite bzw. überregionale zentrale Ansätze zum Energiemanagement, die mit zunehmender Anzahl von Erzeugern erneuerbarer Energie heute jedoch an ihre Grenzen stoßen (vgl. [Leh10, S. 3]). Da die Lasten wie auch die meisten Erzeuger erneuerbarer Energien dezentral lokalisiert sind, sollen in dieser Arbeit insbesondere dezentrale Systeme zum Energiemanagement betrachtet werden. Analog wird in den meisten nachfolgenden Quellen ein zentrales Energiemanagement folglich nur als eine externe Anforderung im Sinne des beschriebenen Demand Responses angesehen.

### 2.1.3 Beweggründe aus Sicht der Endverbraucher

Grundsätzlich ist anzunehmen, dass **Endverbraucher** zuverlässig mit Energie versorgt werden wollen, dafür aus wirtschaftlicher Perspektive aber möglichst wenig vergüten wollen. Daraus kann gefolgert werden, dass Endverbraucher durchaus motiviert sind, entsprechenden Stakeholdern die in Unterabschnitt 2.1.2 beschriebenen Eingriffsmöglichkeiten zu gewähren, sofern sich daraus keine wesentlichen Nachteile, wie Komforteinbußen, ergeben. In Deutschland sind in den letzten Jahren die Energiepreise zudem deutlich angestiegen. Dies lag u. a. an den zusätzlichen gesetzlichen Umlagen und Abgaben, wie z. B. der EEG-Umlage<sup>2</sup> (EEG 2017 § 60), dem KWK-Aufschlag<sup>3</sup> (KWKG §§ 26a und 26b), der Umlage aus StromNEV § 19, der Offshore-Netzumlage (EnWG § 17f), der Umlage für abschaltbare Lasten (AbLaV § 18) sowie an der Stromsteuer (StromStG § 1).

---

<sup>2</sup>siehe auch Erhöhung der EEG-Umlage in [Bun18, S. 28]

<sup>3</sup>KWK steht für Kraft-Wärme-Kopplung

Neben den ökonomischen Anreizen benennt die Quelle [KCM12] für Privatkunden vier weitere Gründe für die Einführung eines Energiemanagements:<sup>4</sup>

1. *Umwelt schonen*: Das Umweltbewusstsein vieler Menschen ist gestiegen. Viele wollen den Ausstoß von CO<sub>2</sub>-Emissionen verringern. Vom Energiemanagement erhoffen sich die Kunden eine Verbrauchsreduktion und damit auch eine Senkung ihrer CO<sub>2</sub>-Emissionen.
2. *Ruhiges Gewissen*: Mittels automatisierter Schaltprogramme können Geräte zeitgesteuert oder z. B. bei Abwesenheit aller Bewohnerinnen und Bewohner abgeschaltet werden. Diese wünschen sich weiterhin, dass ihre Geräte überwacht oder sogar selbstständig abgeschaltet werden können, damit sie keine Gefahren darstellen können. Auch die Abfrage des Status der Geräte aus der Ferne kommt zunehmend in den Fokus.
3. *Komfort*: Durch elektronische Steuerungen können wiederkehrende Abläufe automatisiert werden. Viele Menschen erhoffen sich durch mehr Automatisierung und durch einfachere Bedienungen eine Erhöhung der Lebensqualität. Beliebt sind bereits heute beispielsweise selbstregelnde Lichtsteuerungssysteme, welche die Rollläden sowie die Beleuchtung in Abhängigkeit des Sonnenstands und anwesender Bewohner ansteuern. Mit Umsetzung des Energiemanagements könnten auch solche Zusatzfunktionen realisiert werden.
4. *Gemeinwohl*: Durch die Schonung von Ressourcen sowie durch die Vermeidung des Ausbaus von zusätzlichen Einspeise- und Übertragungskapazitäten kann eine höhere gesellschaftliche Akzeptanz sowie ein höheres wirtschaftliches Vermögen erreicht werden. Auch die Erhaltung der Umwelt, als gemeinsames Gut, tritt hier auch nochmals auf. Die Bereitschaft steigt, mit Hilfe des Energiemanagements zu diesen beiden Aspekten des Gemeinwohls etwas beitragen zu können. (vgl. [KCM12, S. 2f])

### 2.1.4 Maßnahmen des Energiemanagements

Elektrische Geräte nutzen Energie zur Gewährleistung einer Aufgabe. Diese Aufgaben werden in der Regel jedoch nur für oder mit dem Menschen umgesetzt. Teils dienen Geräte auch anderen technischen Einrichtungen, indem sie diesen wiederum einen Dienst bereitstellen. Ein Beispiel wäre hier der WLAN-Router, der anderen Geräten eine funkbasierte Abbindung an das lokale Netzwerk ermöglicht oder der Druckluftkompressor, der anderen Einrichtungen Druckluft zur Verfügung stellt. Das Energiemanagement auf der Verbrauchsseite muss also die Geräte und deren Funktionen berücksichtigen. Nachfolgend soll dargelegt werden, welche Möglichkeiten zum Eingriff ergriffen werden können. Diese werden nachfolgend auch

---

<sup>4</sup>Genauer geht es in der Quelle um den Einsatz von automatisierten Systemen zum Energiemanagement



als *Maßnahmen des Energiemanagements* bezeichnet. Anhand dieser Auflistung soll die Vielfältigkeit der Maßnahmen verdeutlicht werden.

**Gerätetausch:** Wie bereits erwähnt, können im Sinne der Erhöhung der Energieeffizienz Energieeinsparungen und damit auch Kosteneinsparungen durch den Tausch von älteren Geräten durch modernere energiesparendere Geräte bewirkt werden. Entscheidend bei solchen Neuanschaffungen ist eine möglichst kurze Amortisationszeit<sup>5</sup>. Erst nach dieser Zeit wird auch eine Gesamtersparnis erwirtschaftet. Durch staatliche Förderprogramme kann diese Zeit reduziert werden. (vgl. [PD11, S. 381 und 383])

**Versteckte Probleme identifizieren:** Die Quelle [PD11, S. 383] verweist darauf, dass praktisch jede Umgebung versteckte Probleme aufweist, die zu einem erhöhten Energieverbrauch führen. Als Beispiel wird auf defekte Gerätefunktionen und auf falsch konfigurierte Steuerungen verwiesen. Aus industrieller Sicht werden beispielhaft zugesetzte Filter einer Lüftungsanlage und Leckagen in Druckluftsystemen aufgezählt. Die Behebung solcher Probleme führt auch hier zu einer Erhöhung der Energieeffizienz. (vgl. [PD11, S. 383])

Solche Probleme sind für private Nutzerinnen und Nutzer kaum zu erfassen, da sie nur durch aufwändige Einzelmessungen, sofern überhaupt möglich, Einsichten in den elektrischen Energieverbrauch erfahren können. Auch die Identifikation von ineffizienten, technisch veralteten, Geräten ist für Laien häufig nur bedingt möglich. Durch den permanenten Einsatz von Messfunktionen und geeigneten Visualisierungen kann für Nutzerinnen und Nutzer eine höhere Transparenz geschaffen werden. (vgl. [PD11, S. 383])

**Eigene Handlungen anpassen:** Durch einen solchen erweiterten Zugang wäre die interessierte Nutzerin bzw. der interessierte Nutzer auch in der Lage, zu identifizieren, welche ihrer bzw. seiner Handlungen wie viel Energie verbraucht. Entsprechend könnte sie ihre bzw. er seine Handlungen zukünftig anpassen, um den Energieeinsatz zu reduzieren.

**Abstimmung von Geräten:** Nicht zu jeder Zeit wird der Dienst eines vorhandenen Gerätes benötigt. Insbesondere Geräte, die Dienste für andere Geräte bereitstellen, oder Geräte, die einen Dienst für anwesende Bewohner bereitstellen, könnten zeitweise abgeschaltet oder in der Funktion eingeschränkt werden. So muss ein WLAN-Router kein WLAN (mit voller Reichweite) bereitstellen, wenn keiner der Bewohner anwesend ist. Auch besteht die Möglichkeit (elektrische) Heizungen und Lüftungsanlagen bei Abwesenheit zeitweise in ihrer Leistung abzusenken. Zur Ansteuerung können hier schon einfache Zeitschaltuhren genutzt werden. Zu deren Konfiguration muss die Nutzerin bzw. der Nutzer jedoch ein Verständnis von den gegenseitigen Abhängigkeiten der Geräte besitzen.

---

<sup>5</sup>Die Amortisationszeit ist die Zeit, in der sich die zusätzlichen Anschaffungskosten gegen die Energieersparnis rechnen

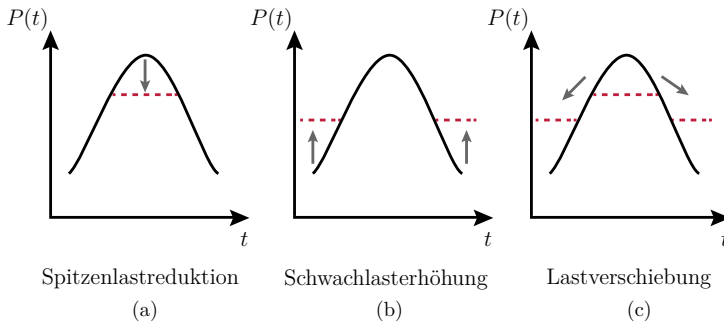


Abbildung 2.2: Maßnahmen des Energiemanagements (a) zur Reduktion, (b) zur Erhöhung und (c) zur Verschiebung des Energieeinsatzes (nach [KCM12, S. 5]).

**Steuerung von Geräten aufgrund von Tarifen und Direkteingriffen:** Endverbraucher mit zeitvariablen Tarifen streben eine zusätzliche Beeinflussung ihrer Geräte an, um weitere Kosten zu sparen. Auch die externen Beeinflussungsmöglichkeiten des Direkteingriffs (beide Arten des Demand Responses) erfordern feingranulare Steuerungsmöglichkeiten.

Die Literatur unterscheidet dabei verschiedene Steuerungseingriffe, die in Abbildung 2.2 dargestellt sind. Als *Spitzenlastreduktion* (engl. Peak Clipping) wird der Vorgang bezeichnet, einen Energieverbraucher zeitweise auszuschalten, um eine ungünstige Verbrauchssituation zu umgehen. Dabei kann es sich beispielsweise um einen Zeitpunkt mit sehr hoher Last, einer sogenannten *Lastspitze*, handeln. Der Dienst oder auch ein Teildienst des Gerätes steht dabei während des Steuerungseingriffs nicht zur Verfügung. Bei der *Lastverschiebung* (engl. Load Shifting) soll ebenfalls eine ungünstige Verbrauchssituation vermieden werden. Hierbei soll der Dienst jedoch vorgezogen oder nachgeholt werden. Handelt es sich um einen kontinuierlichen Dienst, kann der Dienst in Kombination mit einem Speicher (jeglicher Energieform) weiter aufrechterhalten werden. Der Speicher muss folgend (oder zuvor) wieder aufgefüllt werden. Dies betrifft insbesondere Geräte, die mit thermischer Energie arbeiten, wie Kühlschränke oder Heizanlagen. Bei der dritten Variante, der *Schwachlasterhöhung* (engl. Valley Filling) wird hingegen bewusst ein höherer Verbrauch herbeigeführt. Situationen, in denen genügend elektrische Energie zur Verfügung steht, bzw. elektrische Energie günstig ist, werden als Schwachlastphasen bezeichnet. Sie eignen sich, um bewusst Speicher aufzufüllen. In der Vergangenheit wurden so vermehrt Nachtspeicherheizungen eingesetzt. (vgl. [KCM12, S. 4], [Leh10, S. 25], [PRO03, S. 107])

**Steuerung von Geräten aufgrund einer *Lastspitzenreduktion*:** (Industrielle) Großkunden werden bei den Energievertriebsunternehmen in der Regel als Sondervertragskunden geführt und müssen zusätzlich zum Grundpreis einen sogenannten Leistungspreis zahlen.

Der Leistungspreis bezeichnet ein zu entrichtendes Entgelt, welches sich aus der höchsten Viertelstundenlast (Spitzenleistung) eines Jahres berechnet. Durch die Reduzierung dieser Spitzenleistung kann auf das Jahr gerechnet eine hohe Kostenersparnis erlangt werden (vgl. [Kah+12, S. 40]). Entsprechend möchten insbesondere diese Kunden ihren Energieverbrauch durch die zuvor beschriebenen Varianten beeinflussen.

**Steuerung von Geräten bei Prosumern:** Das Ziel der *Eigenverbrauchsoptimierung* bei Prosumern ist es, Geräte im Betrieb so zu beeinflussen, dass die benötigte elektrische Energie möglichst aus der lokalen Energieerzeugung stammt. Sofern möglich, möchten die Prosumer die beschriebenen Steuerungseingriffe nutzen und ihre steuerbaren Energieerzeuger sowie -speicher selbst regeln.

## 2.2 Energiemanagementsysteme und ihre Rahmenbedingungen

Das Energiemanagement unterliegt, wie in Abschnitt 2.1 gezeigt, vielen verschiedenen Beweggründen, die mit unterschiedlichen Maßnahmen bedient werden können. Zu vielen dieser Maßnahmen haben sich Hilfssysteme ausgeprägt, die die Nutzerinnen und Nutzer beim Energiemanagement unterstützen oder dieses überhaupt erst ermöglichen (siehe Definition Energiemanagementsysteme in Kapitel 1). In [PW16, S. 216f] werden zunächst zwei grundlegende Ansätze an *Energiemanagementsystemen* unterschieden:

- *Informationssysteme zur Erhöhung der Verbrauchstransparenz:* Durch die hohe Anzahl an Daten (viele Geräte, lange Zeiträume) ist ein System zur Sammlung und Darstellung von Energieverbrauchsdaten zweckdienlich. Dies erhöht die Verbrauchstransparenz und ermöglicht es den Benutzerinnen und Benutzern, Ineffizienzen zu erkennen. Durch den damit verbundenen Lernprozess sollen die Benutzerinnen und Benutzer in die Lage versetzt werden ihre Geräte selbst effizienter zu nutzen. Diese Systeme werden nachfolgend auch als *Nutzer-unterstützende Energiemanagementsysteme* bezeichnet.
- *Automatisierte Systeme zur Steuerung von Geräten:* Da die automatische Koordination des Energieverbrauchs benachbarter Geräte oder Gerätegruppen möglichst ohne Beteiligung der Nutzerinnen und Nutzer erfolgen soll, wird eine Infrastruktur zur Ansteuerung benötigt. Auch für die automatisierten Eingriffsmöglichkeiten des Lastmanagements (siehe Unterabschnitt 2.1.2) ist eine entsprechende Infrastruktur zur Steuerung der Geräte erforderlich. Diese könnte zum Beispiel aus einem zentralen Gerät

mit einer entsprechenden Energiemanagement-Software sowie einem Kommunikationsnetzwerk zur Anbindung der Geräte bestehen. Solche Systeme werden nachfolgend auch als *automatisierte Energiemanagementsysteme* bezeichnet.

Neben diesen zwei grundlegenden Ansätzen können Energiemanagementsysteme auch weiter unterteilt werden, wie in [PW16, S. 216f] dargestellt.<sup>6</sup>

In den nächsten Unterabschnitten werden verschiedene Systeme nach diesen grundlegenden Ansätzen genauer betrachtet. Dabei wird auf die Systeme für Privathaushalte und für den industriellen Ansatz getrennt eingegangen, da sich in diesen Bereichen verschiedene Herangehensweisen ausgeprägt haben. Wegen der Vielzahl der auf dem Markt bereits verfügbaren Systeme, werden keine konkreten Produkte vorgestellt, sondern es wird auf generelle Systembeschreibungen solcher Systeme zurückgegriffen. Da aussagekräftige Systembeschreibungen von Produkten am Markt schwer zu finden sind, wird auf entsprechende Beschreibungen aus der Fachliteratur zurückgegriffen. Hierbei wird davon ausgegangen, dass sich der Systemaufbau von Produkten am Markt mit den Systemen aus der Literatur ähnelt. Betrachtungen des Marktes konnten dieses nicht widerlegen. Neben den Systembeschreibungen gibt es weitere Ansätze zur Verallgemeinerung, wie z. B. Anstrengungen zu Standardisierungen (siehe u. a. [SCN15]), die jedoch hier nicht herangezogen werden, da sie als nicht zielführend für diese Arbeit erachtet wurden.<sup>7</sup>

In Unterabschnitt 2.2.1 wird anhand einer breiten Literaturlauswahl ein Gesamtbild über automatisierte Energiemanagementsysteme in Privathaushalten vorgestellt, um den aktuellen Stand der Forschung zu verdeutlichen. Die Quellen folgen dabei durchgehend einer ähnlichen Architektur, die zunächst vorgestellt wird. Die Systeme selbst fokussieren auf unterschiedliche Optimierungsziele und verwenden unterschiedliche Techniken zur Einbindung von Geräten. Darauf aufbauend werden in Unterabschnitt 2.2.2 die automatisierte Energieverbrauchsoptimierung und die dazu notwendigen Optimierungsalgorithmen dieser Systeme beleuchtet. Auch hier wird eine breite Literaturlauswahl herangezogen, um die große Anzahl unterschiedlicher Optimierungsalgorithmen zu verdeutlichen. Anschließend werden verschiedene Ansätze für Energie-Informationssysteme für Privathaushalte in Unterabschnitt 2.2.3 vorgestellt. Energiemanagementsysteme für den industriellen Einsatz werden in Unterabschnitt 2.2.4 beschrieben.

Aus den Beschreibungen dieser Unterabschnitte wird deutlich, dass automatisierte Energiemanagementsysteme und Energie-Informationssysteme unterschiedliche Ziele verfolgen,

---

<sup>6</sup>In dieser Arbeit sollen die Systeme jedoch nicht weiter unterschieden, sondern vielmehr die verschiedenen Systeme zusammengeführt werden.

<sup>7</sup>[SCN15] behandelt lediglich den Informationsaustausch zwischen dem Energiemanagementsystem und Geräten.

weshalb sie in der Literatur überwiegend getrennt betrachtet werden. Dennoch arbeiten die Systeme auf Grundlage der gleichen Geräte. Entsprechend könnten sich Synergieeffekte ergeben, wenn die Systeme und ihre Ansätze nicht weiter getrennt betrachtet würden. Auch fällt die Aufteilung in Privathaushalte und Unternehmen auf, obwohl auch hier Parallelen und potentielle Synergieeffekte deutlich werden.

### 2.2.1 Automatisierte Energiemanagementsysteme für Privathaushalte

In der Literatur sind Energiemanagementsysteme für den Privatgebrauch unter einer Vielzahl von Bezeichnungen zu finden. Die Quelle [BZ15] listet bereits 31 Namen für solche Systeme auf. Am Häufigsten werden die englischen Begriffe *Home Energy Management System* (engl., HEMS) oder allgemeiner *Energy Management System* (engl., EMS) oder auch *Smart Home Energy Management* (engl., SHEM) verwendet. Diese Systeme setzen den in Unterabschnitt 2.1.4 beschriebenen Ansatz der automatisierten Steuerung von benachbarten Geräten um.

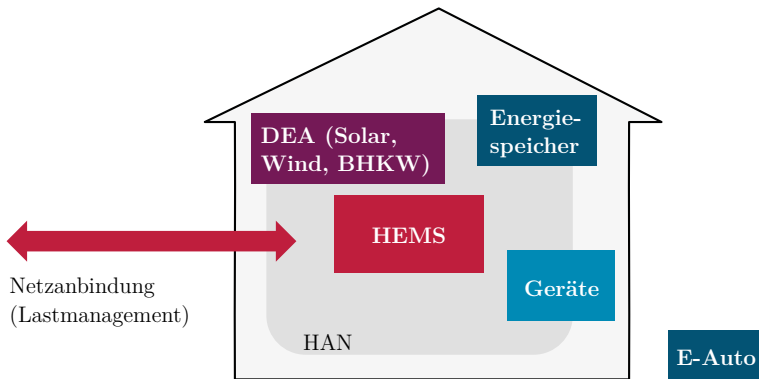


Abbildung 2.3: HEMS-Architektur mit Geräten in Hellblau, dezentralen Energieanlagen (DEA) in Lila sowie Energiespeichern in Dunkelblau. Durch eine Kommunikationsanbindung ist auch eine Beeinflussung durch externe Parteien möglich (in Anlehnung an [KCM12, Abbildungen 1 und 8], [PW16, Abbildung 1] und [Sha+18, Abbildung 1])

Die Quellen [KCM12], [PW16] und [Sha+18] beschreiben jeweils die allgemeine Architektur des HEMS. Abbildung 2.3 zeigt eine kombinierte Darstellung der in den Literaturquellen erläuterten Konzepte. Abgebildet ist ein symbolischer Privathaushalt mit Geräten, Erzeugern erneuerbarer Energie sowie dezentralen Speichern. Weiterhin ist ein Elektroauto abgebildet, dass als Energiespeicher betrachtet wird. Das HEMS ist die zentrale Schaltstelle für das automatisierte Energiemanagement aller lokalen Energieverbraucher und -erzeuger.

Steuerungssignale für Direkteingriffe oder zur Kommunikation von variablen Tarifen (siehe Unterabschnitt 2.1.2) werden über eine Kommunikationsanbindung an das Energieversorgungsnetz umgesetzt. Als Kommunikations-Gateway wird oft der *intelligenter Stromzähler* (engl. Smart Meter) vorgesehen.

Das HEMS selbst ist in einem Netzwerk, dem *HAN* (engl. Home Area Network) lokalisiert. Im Gegensatz zum *lokalen Netzwerk* (engl. Local Area Network, LAN), das den lokalen (aber nicht begrenzten) Zusammenschluss von IT-Geräten bildet, beschreibt das HAN den Zusammenschluss *aller* Geräte mit Kommunikationsschnittstelle innerhalb eines Privathaushaltes. Die Geräte stellen hierzu dem HEMS ihre Mess- und Steuerungsfunktionalitäten über das HAN bereit. Das HAN kann auch nicht-IP-basierte Kommunikationsmedien, wie beispielsweise Feldbus-Systeme, umfassen. Neben leitungsgebundenen Kommunikationsmedien kommen auch funkbasierte Kommunikationsstandards zum Einsatz. (vgl. [KCM12, S. 2 und S. 5-7])

Die hier beschriebene vereinfachte allgemeine Architektur wird in einer größeren Anzahl von Literaturquellen herangezogen<sup>8</sup>. So finden sich oft vergleichbare Schaubilder und Beschreibungen (siehe z. B. [Zha+19, S. 1392], [Cel+18, S. 6389], [CC18, S. 2], [Li+18a, S. 2-3], [AS18, S. 1330-1331], u. v. m.). Die Mehrzahl der betrachteten Quellen hat das Ziel, eine Eigenverbrauchsoptimierung (siehe Unterabschnitt 4.1.4) mit Hilfe von Optimierungsalgorithmen auf Basis von variablen Tarifen (siehe Unterabschnitt 4.1.2) durchzuführen. In anderen Quellen wird das Ziel der Energieeinsparung formuliert, indem Abhängigkeiten zwischen Geräten sowie den Nutzerinnen und Nutzer modelliert (siehe auch Unterabschnitt 2.1.4) und die Geräte entsprechend angesteuert werden (so z. B. [NDM18]). Wenige Quellen mit Bezug zum automatisierten Energiemanagement setzen das Ziel eines Direkteingriffs um (so z. B. [KKA19], [Vij+18]).

Für die Kommunikationsanbindung der Geräte über das HAN werden in vielen Quellen mehrere Kommunikationsstandards und -systeme benannt. So werden unter anderem ZigBee, Z-Wave, KNX, X-10, BACnet, Bluetooth, WLAN, GSM und 3G (vgl. [Mah+15, Seite 1440], [ZC18, S. 1] und [Kri+18, S. 1] aufgezählt. Konkrete Umsetzungen finden sich unter anderem auf Basis von ZigBee (siehe z. B. [Zho+16], [Sow+18, S. 3]), Bluetooth (siehe z. B. [AB18, S. 20]) und LAN/WLAN.

Um herkömmliche Geräte einzubinden, setzen einige Literaturstellen (z. B. [Zha+19, S. 1392], [Li+18b, S. 2216-2217]) schaltbare Zwischenstecker mit Kommunikationsanbindung und Energiemessfunktion ein. Solche Zwischensteckdosen werden auch als *Smartplugs* bezeichnet.

---

<sup>8</sup>Zugrunde liegt eine Literaturrecherche aktueller Quellen zum Thema Energiemanagementsysteme für Privathaushalte

Die überwiegende Anzahl der betrachteten Quellen setzt die Gerätekommunikation lediglich voraus und bedient sich aufgenommenen Datensätzen (wie REDD [KJ11], dem Smart Grid Smart City Dataset [MFG15, S. 192ff]) oder simulierter Geräte. (so z. B. [Dao+18], [Cel+18, S. 6390], [CC18, S. 3], [Awa+18, S. 2-3], [Awa+18, S. 1277], [AS18, S. 1336], [Luo+19, S. 2 und S. 6-7], [Ant+18, S. 3]) Diese Quellen verweisen für die Geräteanbindung vielfach auf erweiterte Themengebiete, wie das Smart Home und das Internet der Dinge (siehe Kapitel 1). Diese Themengebiete werden gemeinsam mit konkreten Systemen erst in Abschnitt 2.3 vorgestellt.

Laut [BZ15, S. 320] können mit der Realisierung solcher Systeme Kosteneinsparungen von 23,1 % für elektrische Energie bzw. von 29,6 % der Spitzenleistung gemittelt aus 25 bzw. 18 Literaturstellen nachgewiesen werden.

Zusammengefasst werden mit dem HEMS bereits verschiedene Beweggründe aus Unterabschnitt 2.1.2 und 2.1.3 adressiert. In der Regel wird von einer Quelle jedoch nur eine Maßnahme zum Energiemanagement (siehe Unterabschnitt 2.1.4) bedient. Auch gibt es keine einheitliche Vorgehensweise, mit Geräten zu kommunizieren. Bedingt durch die große Anzahl der bereits verfügbaren Kommunikationsstandards kann gefolgert werden, dass heterogene Lösungen benötigt werden, die mehrere Kommunikationsstandards ansprechen.

In [Xie+16] wird als Erweiterung des HEMS das Community Energy Management System (engl. für Gemeinschafts-Energiemanagementsystem, CEMS) vorgeschlagen. Die Grundidee ist die Ausweitung der automatisierten Energieverbrauchsoptimierung auf größere Bereiche, wie Häuser, Straßen, Quartiere oder sogar ganze Städte, wie bereits in Unterabschnitt 2.1.2 benannt. Als Beispiel wurden in der Quelle Überspannungen kompensiert, die durch erneuerbare Energieerzeuger im Verteilnetz hervorgerufen werden. Das CEMS bildet dabei ein Bindeglied zwischen lokalen Systemen, wie dem HEMS und übergeordneten Systemen. (vgl. [Xie+16, S. 224 und S. 229])

## 2.2.2 Algorithmen zur automatisierten Energieverbrauchsoptimierung

Die Kernaufgabe des HEMS (siehe Unterabschnitt 2.2.1) ist die *automatisierte Energieverbrauchsoptimierung*. Diese hat das Ziel, mittels eines Optimierungsalgorithmus eine Ablaufplanung (Scheduling) von benachbarten Geräten, Energieerzeugern und -speichern zu erstellen, sodass ein oder mehrere Optimierungsziele erreicht werden können. Schematisch ist dieser Aufbau in Abbildung 2.4 in Form eines Regelkreises dargestellt. Der Regler (hier Optimierungsalgorithmus) hat die Aufgabe, die Regelstrecke (hier die Geräte) zu beeinflussen, um bestimmte Werte für die Regelgrößen (hier bspw. den Energieverbrauch) zu erreichen. Diese Darstellungsform dient als Ersatz für jegliche Arten von Algorithmen, zeigt

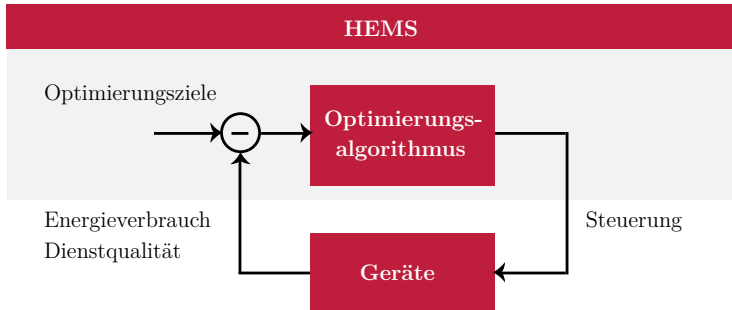


Abbildung 2.4: Verallgemeinerter algorithmischer Ablauf eines HEMS, hier dargestellt in Form eines Reglers. Der Energieverbrauch und die Dienstqualität sind hier exemplarisch als Regelgrößen eingefügt. Viele Algorithmen arbeiten im Sinne einer Ablaufplanung (Scheduling). Üblicherweise handelt es sich bei den Regelgrößen um Pläne und Prognosen der Größen. (eigene Darstellung)

aber repräsentativ die Abhängigkeiten der Größen. Der Optimierungsalgorithmus wird in regelmäßigen Abständen ausgeführt, um die Ablaufpläne zu aktualisieren bzw. die Energieerzeuger und -speicher einzustellen. Die Optimierung findet dabei üblicherweise auf Grundlage von prognostizierten Größen statt, wie beispielsweise auf der Basis des prognostizierten Energieverbrauchs.

In der Literatur finden sich eine Vielzahl verschiedener Optimierungsziele und Regelgrößen nach denen Geräte optimiert werden können. Wie bereits in Unterabschnitt 2.2.1 beschrieben, fokussieren die meisten Quellen auf die Optimierung des Eigenverbrauchs. Dabei werden je nach Quelle unterschiedliche variable Tarife (TOU, RTP und CPP, siehe Unterabschnitt 2.1.2) zugrunde gelegt.

In einigen Quellen werden neben den Geräten auch dezentrale Energieerzeugungsanlagen modelliert und mit optimiert. Auch Energiespeicher werden teilweise eingebunden. In einigen Quellen wird auch die Integration von Elektrofahrzeugen untersucht. (siehe z. B. [Ebr+18], [Wu+18], [CC18], [AS18]).

Als Regel- bzw. Einflussgröße wird in diesen Quellen der Energieverbrauch genutzt. Diese Größen werden mittels der variablen Tarife als Kosten modelliert. Zusätzlich wird entweder eine Dienstqualität als zusätzliche Größe eingebracht, oder es werden Rahmenbedingungen definiert, die eine Geräteaktivität erzwingen. Andere Verfahren verfolgen das Ziel, zusätzlich die Unannehmlichkeiten (engl. inconvenience) oder die Qualität der Erfahrung (engl. Quality of Experience, QoE) der Bewohner zu modellieren (siehe z. B. [Pil+18], [Ebr+18], [Dol+18], [BZ15, S. 232ff]). In [Cha+19] wird der Datenschutz als weitere Regelgröße modelliert, um zwischen Kosteneinsparungen und Privatsphäre abwägen zu können.



Für alle diese Optimierungsziele kommen sehr verschiedene Arten an Algorithmen zum Einsatz. In [Sha+18, S. 24504] werden drei Arten unterschieden: Regelbasierte Verfahren, Verfahren unter Verwendung von künstlicher Intelligenz und Optimierungsverfahren.

Bei **regelbasierten Verfahren** werden anhand von einfachen Bedingungen Abhängigkeiten geschaffen. Durch unterschiedliche Priorisierung können Geräte zusätzlich stufenweise nach und nach ab- bzw. zugeschaltet werden. Der Nachteil dabei ist die schlechte Erweiterbarkeit der Regeln bei der Hinzunahme weiterer Geräte. (vgl. [Sha+18, S. 24504]) Solche Verfahren werden beispielsweise in [Vij+18], [ZC18] und [Kri+18] verwendet.

In der Kategorie der **Optimierungsverfahren** entstammen viele der Algorithmen der mathematischen Optimierung, wie die lineare, die quadratische und die dynamische Optimierung. Implementierungen finden sich u. a. auf Basis der gemischten ganzzahligen linearen Optimierung (engl. Mixed Integer Linear Programming, siehe z. B. [Ant+18, S. 2]) und der approximativen dynamischen Optimierung (engl. Approximate dynamic programming, ADP, siehe z. B. [KVC18]).

Auch heuristische Verfahren – also Optimierungsverfahren speziell für einen Einsatzzweck – werden mit Erfolg eingesetzt. Dabei handelt es sich in der Regel um wissensbasierte Lösungen, die in Form von Regelsätzen beschrieben werden. Die Regelsätze müssen zunächst von Experten erstellt werden. Sie besitzen im Vergleich zu den Algorithmen der mathematischen Optimierung einen sehr geringen Rechenaufwand. (vgl. [BZ15, S. 329]) Implementierungen finden sich u. a. für den genetischen Algorithmus (engl. Genetic Algorithm, GA) (siehe z. B. [Awa+18], [AS18], [Rah+16, S. 461-462]), die binäre Partikelschwarmoptimierung (engl. Binary Particle Swarm Optimization, BPSO) (siehe z. B. [Rah+16, S. 462]) und den Ameisenalgorithmus (engl. Ant Colony Optimization, ACO) (siehe z. B. [Rah+16, S. 463]).

Auch stochastische Verfahren (siehe z. B. [Wu+18], [YHS18]) sind verbreitet, die auf Grundlage großer Datenmengen arbeiten. Weiterhin werden auch Markoff-Entscheidungsprozesse verwendet (z. B. in [Awa+18], [Han+18]).

Verfahren unter Verwendung von **künstlicher Intelligenz** nutzen z. B. künstliche neuronale Netze (engl. Artificial Neural Network, ANN) um zu erlernen, welches der beste Prozessablauf für ein Gerät ist (siehe z. B. [Tas+15]). [Sha+18] zählt auch die Fuzzy-Logik-Steuerung (engl. Fuzzy Logic Control, FLC) zu dieser Kategorie. Implementierungen finden sich z. B. in [PG18] und [Sou+18].

Für viele der Algorithmen muss das Verhalten der Geräte bekannt bzw. vorhersagbar sein. Dazu nutzen viele Literaturquellen Modelle, die das Verhalten der Geräte nachbilden. Jedoch unterscheiden sich Geräte teilweise deutlich voneinander. Um nicht für jedes Gerät ein eigenes Modell zu erarbeiten, haben verschiedene Forschergruppen Geräte anhand des

Energieverbrauchs eingeordnet und entsprechende allgemeine Gerätegruppen vorgeschlagen (siehe z. B. [BZ15, S. 233]).

Auf dem Markt gibt es bereits erste konkrete Produkte für Privatanwenderinnen und -anwender, um eine Optimierung der Betriebszeiten von Geräten in Abhängigkeit einer Solaranlage und Batteriespeichern zu ermöglichen. Den Systembeschreibungen nach handelt es sich im Vergleich zu den oben genannten Quellen dabei überwiegend um rudimentäre regelbasierte Systeme.

### 2.2.3 Energie-Informationssysteme

Neben den automatisierten Systemen finden sich in der Literatur auch Systeme für Privathaushalte, die dem Ansatz der Informationssysteme zuzuordnen sind. Diese werden teilweise als *Energy Information System* (engl., EIS) bezeichnet. Typische Bestandteile dieser Systeme sind: (vgl. [PD11, S. 383] sowie [PW16, S. 217])

- Infrastruktur zur Datenerfassung (Sensornetzwerk, Datenlogger, ..),
- Applikationsserver mit Datenbank (inkl. Berechnungs- und Analyse-Algorithmen, Benachrichtigungsdienst, Reporting), sowie
- Nutzerinterface zur Visualisierung und Konfiguration (z. B. In-Home Displays (IHD), Tablets).

Diese Arten der Systeme sollen den Nutzerinnen und Nutzern Feedbackmöglichkeiten über Anomalitäten in ihren Haushalten anbieten. Klassische Analysendarstellungen sind nach [PD11, S. 383]:

- Auftragung des Grundverbrauchs gegen den Spitzenverbrauch, um den Energieanteil ohne Nutzung identifizieren zu können (so z. B. Standby-Verbräuche),
- Zeitbereichsdarstellung des Verbrauchs bezogen auf eine Woche, um versteckte Verbraucher in der Nacht oder am Wochenende erkennen zu können,
- Benchmarks als Vergleichsmöglichkeit zu anderen, sowie
- Korrelationsanalysen, um Abhängigkeiten z. B. von der Außentemperatur oder der Sonneneinstrahlung festzustellen.

Mit dem Einsatz dieser Systeme sollen Anwenderinnen und Anwender in die Lage versetzt werden, selbst den Energieverbrauch ihrer Geräte nachvollziehen zu können. Dadurch können sie ggfs. ihr Verhalten anpassen oder sogar selbst Energiespar-Maßnahmen identifizieren, wie z. B. den Austausch ineffizienter Geräte (siehe Unterabschnitt 2.1.4).

Veröffentlichungen zu diesem Themengebiet beziehen sich beispielweise auch auf die Art und das Design der Rückmeldungen (engl. Feedback) (siehe z. B. [Bar15]).

Weiterhin gibt es viele Quellen, die sich mit Systemen zur *Gerätedisaggregation* beschäftigen (für einen Überblick siehe [KK16]). Die Technik wird im englischen als *Nonintrusive Load Monitoring* (engl., NILM) oder auch *Nonintrusive Appliance Load Monitoring* (engl., NIALM) bezeichnet. Ihr Ziel ist es, aus dem aggregierten Energieverbrauch mehrerer Geräte den Energieverbrauch bzw. Betriebsstatus der einzelnen Geräte zu ermitteln und den Nutzerinnen und Nutzern darzustellen. Entsprechend wird diese Technik hier ebenfalls den Informationssystemen zugeordnet. Für die Technik kommen u. a. automatische Lernverfahren, wie künstliche neuronale Netze oder *Hidden-Markoff-Modelle* (engl. Hidden Markov Model, HMM) zum Einsatz (siehe z. B. [Zoh+13], [WLR18]).

## 2.2.4 Energiemanagementsysteme für Unternehmen und Einrichtungen

Das Ziel des Energiemanagements im Kontext von Unternehmen entspricht weitestgehend der in Kapitel 1 benannten Definition. Dabei soll der Energieeinsatz in industriellen und gebäudetechnischen Anlagen sowie in Büros durch Maßnahmen optimiert werden (vgl. [Kah+12, S. 16]). Für Unternehmen und Einrichtungen besteht bisher keine Pflicht zur Durchführung eines Energiemanagements. Jedoch ist die Zertifizierung eines Energiemanagements inzwischen Voraussetzung, um an bestimmten Fördermaßnahmen teilnehmen zu können. Die Zertifizierung erfolgt dabei nach Norm DIN EN ISO 50001. (vgl. [Kah+12, S. 18])

Im Kontext von Unternehmen wird unter dem Begriff eines Managementsystems die systematische betriebsinterne Regelung von Abläufen und Zuständigkeiten verstanden. Das Ziel ist es, die Umsetzung von Unternehmenszielen sicherzustellen. Die Grundlagen solcher Managementsysteme sind in der Regel in entsprechenden Normen hinterlegt und folgen dem PDCA-Kreislauf. Dieser steht für die Worte planen (engl. plan), umsetzen (engl. do), kontrollieren (engl. check) und handeln (engl. act), mit denen die einzelnen Phasen eines fortlaufenden Managements beschrieben werden. Der Prozess wiederholt sich, wobei die Ergebnisse des letzten Durchlaufs als Ausgangslage für den neuen Durchlauf gesetzt werden. Dadurch soll eine fortlaufende Verbesserung von Systemen und Prozessen erreicht werden, da der Prozess eine wiederholende Neubewertung erlaubt. (vgl. [Kah+12, S. 18ff])

Dies gilt auch für das Energiemanagement nach der Norm DIN EN ISO 50001. Dabei werden im Sinne des *PDCA-Kreislaufs* unter *Planen* jegliche Vorbereitungen verstanden, darunter

beispielsweise die Definition unternehmensweiter Energiesparziele sowie die Benennung von Verantwortlichen und die Festlegung von Maßnahmen. Demnach sollte im Unternehmen eine Energiemanagerin oder einen Energiemanager benannt werden, die bzw. der fortan Maßnahmen zur Optimierung des Energieverbrauchs ausarbeitet. Diese Maßnahmen zu priorisieren, in Arbeitspläne zu überführen und deren Realisierung zu überwachen, wird unter der Phase *Umsetzen* verstanden. Bei der Phase *Kontrollieren* soll überprüft werden, ob und inwieweit die festgelegten Energiesparziele erreicht werden konnten. Im letzten Schritt dem *Handeln* sollen die Ergebnisse zusammengefasst und neue Ziele abgeleitet werden. (vgl. [Kah+12, S. 21 und S. 47f])

Nach [Kah+12, S. 16] wird unter dem Begriff *Energiemanagementsystem* im industriellen Kontext ein Unterstützungssystem verstanden, dass die systematische Erfassung der Energieverbräuche ermöglicht und damit als Grundlage für Investitionsentscheidungen dient. Nach der Einteilung der Ansätze von Energiemanagementsystemen in Abschnitt 2.2 zählen diese Systeme zum Bereich der Informationssysteme. Sie unterscheidet sich damit grundsätzlich von den HEMS für Privathaushalte (siehe Unterabschnitt 2.2.1). Maßnahmen, die eine automatisierte Steuerung von Geräten vorsehen, werden in Industrieunternehmen oft auf Basis der vorhandenen Industrie- oder Gebäudeautomatisierung konzipiert und umgesetzt.

Das Bundesamt für Wirtschaft und Ausfuhrkontrolle hat eine Liste mit 217 förderungsfähigen Energiemanagementsystemen herausgegeben (siehe [Bun16]). Diese Literaturstelle definiert dabei Mindestanforderungen, auf die die Systeme überprüft wurden. Demnach muss ein Energiemanagementsystem eine Datenauswertung und Visualisierung aufweisen. Die Systeme müssen zudem einfache Rechenoperationen (Summen, Mittel- und Extremwerte) sowie energetische Kennwerte berechnen und die Daten nach Wunsch der Nutzerinnen und Nutzer anzeigen können. Auch ein Berichtswesen sowie Möglichkeiten zur Alarmierung sollen die Systeme bereitstellen. Bei der Auswahl wurden sowohl die Integrationsfähigkeit in bestehende Systeme sowie der angebotene Support überprüft. (vgl. [Bun16, S. 3])

Viele dieser Energiemanagementsysteme bauen auf einer Infrastruktur mit Energiezählern auf. Zur Datenbereitstellung werden dabei Unternehmensbereiche, Gebäude, Flure, Räume oder sogar einzelne Maschinen separat vermessen. Je kleiner die vermessenen Bereiche sind, desto einfacher können Folgerungen getroffen werden. [Pos16] definiert dazu ein Maß für die Transparenz des Energieverbrauchs.

Neben der oben beschriebenen Norm gibt es weitere Normen, die das Energiemanagement in Unternehmen betreffen, wie zum Beispiel die Norm zu „Energiekenngrößen“ (VDI 4661) und die Norm zur „Energieberatung für Industrie und Gewerbe“ (VDI 3922). Diese müssen mit Hilfe von Energiemanagementsystemen umgesetzt werden. Weiterhin müssen entsprechende Gesetze, wie z. B. die Energieeinsparverordnung (EnEV) beim Energiemanagement in der

Industrie beachtet werden. Die genannte Energieeinsparverordnung fokussiert auf Gebäude und energieeinsparenden Wärmeschutz (siehe EnEV § 1).

## 2.3 Kommunikationsstrukturen für das Internet der Dinge

Neben dem Energiemanagement beschäftigen sich auch andere Themenfelder mit der Kommunikation von Anlagen, Maschinen und Geräten. Darunter sind beispielsweise aufzuführen: (siehe auch [KSG18, S. 4], [RSS15])

- das Themenfeld des *intelligenten Zuhauses* (engl. Smart Home) mit seinen Komfortfunktionen sowie Funktionalitäten zur Fernsteuerung und zur Fernkonfiguration von Geräten (inkl. Beleuchtungssteuerung, Temperaturregelung, ...),
- die *Industriautomation* zur Verknüpfung und Überwachung industrieller Fertigungsanlagen und Werkzeugmaschinen,
- die *Gebäudeautomation, Gebäudeleittechnik* (GLT) bzw. *intelligente Gebäudesteuerung* (engl. Smart Building Control) von gebäudetechnischen Anlagen und Geräten in Abhängigkeit der Anwesenheit von Bewohnerinnen und Bewohner, des Wetters und weiterer Umweltfaktoren,
- die *Sicherheitsüberwachung* von Räumen mittels Sensoren bei Abwesenheit der Nutzerinnen und Nutzer,
- die Steuerung und Verknüpfung von *Multimedia-Geräten*, wie z. B. Musikanlagen und TV-Geräten,
- der Analyse von Maschinen mittels der *prädiktiven Instandhaltung*, um frühzeitig Ausfälle zu erkennen und diesen vorbeugen zu können, sowie
- das Themengebiet des *altersgerechten bzw. umgebungsunterstützten Wohnens* (engl. Ambient Assisted Living, AAL), das z. B. älteren Menschen ermöglichen soll länger selbstständig in ihrer gewohnten Umgebung leben zu können.

Auf einige dieser und weiterer Themenfelder wird aufgrund ihrer Schnittmengen zum Energiemanagement in Unterabschnitt 2.3.1 näher eingegangen. Zusätzlich werden Kommunikationsstandards und -systeme benannt, die eine Anbindung von Geräten ermöglichen.

Im Gegensatz zu den anderen benannten Anwendungsfeldern entsteht für die Bewohnerinnen und Bewohner bzw. Mitarbeiterinnen und Mitarbeiter beim Energiemanagement in der Regel kein direkter Nutzen, wie ein Komfortgewinn. Der Beweggrund für eine Anschaffung

eines Energiemanagementsystems ist deshalb vorrangig eine mögliche Kostenersparnis. Deshalb müssen die potentiellen Energiekostenreduzierungen die Anschaffungskosten und Betriebskosten innerhalb eines annehmbaren Zeitraums übersteigen. Dies gilt sowohl für Privathaushalte als auch für Unternehmen, die ihr Energiemanagement in einem über die gesetzlichen Vorgaben hinausgehenden Umfang betreiben möchten.

Bei einer mehrfachen Nutzung der Infrastrukturen für die benannten Anwendungsfelder ließen sich Synergieeffekte ausnutzen. Insbesondere das Energiemanagement könnte von einer mehrfachen Nutzung der Infrastrukturen profitieren, da die zusätzlichen Anschaffungskosten und Betriebskosten für das Energiemanagement deutlich geringer ausfallen würden.

Das *Internet der Dinge* beschreibt die Idee, dass prinzipiell jeder Gegenstand (an irgendeinem Ort zu irgendeiner Zeit<sup>9</sup>) über das Internet mit anderen Geräten und Gegenständen interagieren kann. (vgl. [MBR15, S. 72-74]) Da das Internet der Dinge bewusst an kein Anwendungsfeld gebunden wurde, ist es gut geeignet zur Zusammenführung mehrerer Anwendungsfelder. Synergieeffekte entstehen dabei nicht nur durch die doppelte Nutzung der Infrastrukturen, sondern auch indem Sensoren, Geräte und sonstige Gegenstände mehrfach, von verschiedenen Anwendungsfeldern verwendet werden können. Die Kernaufgabe ist also die Herstellung einer umfassenden *Interoperabilität* zwischen verschiedenen Dingen, die dann miteinander interagieren sollen. Interoperabilität kann dabei auf verschiedenen Ebenen hergestellt werden.

Beim Internet der Dinge werden mit großer Wahrscheinlichkeit Daten von Gegenständen und Geräten anfallen, die Rückschlüsse auf die Handlungen und Gewohnheiten ihrer Eigentümer erlauben. Mit der Einführung der Übertragung dieser Informationen über das Internet an andere Parteien sind die *Datensicherheit* und der *Datenschutz* ein dringend zu beachtendes Thema. (vgl. [Sta14, S. 6-7]) [Err+18, S. 4381] weist daraufhin, dass dieses Thema trotz der hohen Bedeutung bisher wenig beachtet wurde.

Aus Literaturquellen, die eine Übersicht über das Thema geben, wie [Ray18], und Veröffentlichungen, die eine Definition des Begriffs des Internet der Dinge suchen, wie [MBR15], wird deutlich, dass das Internet der Dinge bisher keine einheitlichen Techniken und Strukturen besitzt. Dagegen finden sich viele verschiedene übliche Lösungen. Aus diesem Grund werden mögliche Kommunikationsstrukturen in Unterabschnitt 2.3.2, Middlewares für die Anbindung von verteilten Geräten in Unterabschnitt 2.3.3 und übliche Protokolle für das Internet der Dinge in Unterabschnitt 2.3.4 vorgestellt.

Abschließend wird in Unterabschnitt 2.3.5 anhand von Fachartikeln der Stand der Forschung in der Fusion der Themen Energiemanagementsysteme und Internet der Dinge vorgestellt.

---

<sup>9</sup>nicht immer und überall, sondern wenn es notwendig und sinnvoll ist

Deutlich wird, dass durch die Zusammenführung der Themen die gewünschten Synergieeffekte ermöglicht werden könnten.

### 2.3.1 Verwandte Infrastrukturen und Systeme

Im Folgenden sollen Infrastrukturen und Systeme beschrieben werden, die entweder ähnliche Umgebungen wie Energiemanagementsysteme bedienen oder an den Schnittstellen zu diesen operieren.

Unter dem Begriff *Smart Home* (engl. für intelligentes Zuhause) können sehr verschiedene Konzepte und Systeme verstanden werden. Nach [Dar18] stehen dabei entweder die Nutzerinnen und Nutzer mit Diensten zur Erhöhung ihres Komforts im Zentrum oder der Fokus liegt auf dem Gebäudemanagement und dessen Optimierung. Die Konzepte verfolgen dabei in der Regel die mögliche Anbindung von Kommunikationsnetzwerken zur Verknüpfung der Geräte oder Anlagen. Ziel ist die Bereitstellung von Diensten, die auf die vorhandenen Fernzugriffe der Geräte aufsetzen. (vgl. [Dar18, S. 140])

Insbesondere im Bereich des Smart Homes sind bei einigen Plattformen Problematiken bei der Vernetzung von Geräten offenkundig geworden. Nach [CD17, S. 47] unterliegt die Heimautomation noch folgenden Problemen bzw. Herausforderungen:

- Unterschiedlichkeit der intelligenten Geräte
- Keine Interoperabilität von Diensten
- Abhängigkeit von zentralisierten Komponenten in der Cloud
- Nichtbeachtung von Sicherheits- und Privatsphäre-Anforderungen von Nutzerinnen und Nutzern
- Unterschiedliche IoT-Standards

Die Quelle folgert, dass diese Probleme zu einer Zersplitterung des Marktes von Smart-Home-Produkten und zu einer Minderung der Verbrauchererfahrung führen. (vgl. [CD17, S. 47])

Mit dem Forschungsbereich der sogenannten *Industrie 4.0* werden auch im industriellen Umfeld Kommunikationslösungen eingesetzt um mehr Intelligenz in den jeweiligen Umgebungen zu ermöglichen. Dabei wird mehr und mehr von der klassischen Industrieautomatisierung abgewichen und Internet-Technik eingesetzt. Vernetzte Maschinen werden in diesem Kontext auch als *Cyber-physische Systeme* (engl. Cyber Physical Systems, CPS) bezeichnet. (vgl. [BHV14, S. 26 und S. 30])

Weiterhin beschäftigen sich verschiedene Themengebieten mit der Vernetzung größerer Bereiche. Hier ist auf das Themengebiet der intelligenten Energieversorgungsnetze (engl. Smart Grid) zu verweisen, dass bereits in Unterabschnitt 2.1.2 angeführt wurde. So sehen auch einige Quellen eine Verknüpfung zwischen Energiemanagementsystemen und dem Smart Grid (siehe z. B. [AB18, S. 21], [Zho+16, S. 31 und S. 33])

Ähnliche Verknüpfungen sind auch zu anderen Themenbereichen denkbar. Die *Smart City* (engl. für intelligente Stadt) ist der Ansatz Aspekte des Managements von Städten mit Hilfe von Informations- und Kommunikationstechnik miteinander zu kombinieren. (vgl. [Mon15, S. 2]) Eine Fassade der Smart City ist das Energiemanagement (aller Energieformen) sowie das Management weiterer Ressourcen (wie Wasser, Abwasser, Müll, ...) der Komponenten einer Stadt. (vgl. [BF17, S. 54ff] sowie beispielhaft [Eja+17, S. 86])

Wie in Unterabschnitt 2.2.1 zusammengefasst, gibt es viele Kommunikationsstandards und -systeme, jedoch bisher keine einheitliche Vorgehensweise zur Kommunikation mit Geräten. Grundsätzlich finden sich drei übliche Kommunikationsmedien, über die eine Kommunikation mit Geräten ermöglicht werden kann. Zu beachten ist, dass die verschiedenen Kommunikationsmedien unterschiedliche Vor- und Nachteile aufweisen. Mittels einer separaten Verkabelung könnten die Geräte direkt angesprochen werden. Hier sind verschiedene Typologien, wie Ringleitungen, Sternverkabelung usw. möglich. Diese Variante ist aufwendig und damit kostenintensiv. Dieser Weg wird entsprechend nur a) bei zeitkritischen Übertragungen, b) in sicherheitskritischen Bereichen und c) ggfs. bei Neubauten vorgenommen. Als weiterer Lösungsansatz können mittels funkbasierter Kommunikation Verbindungen mit geringem Aufwand realisiert werden. Durch die Realisierung mit einem geteilten Kommunikationskanal müssen jedoch Anforderungen hinsichtlich der Sicherheit, der Zuverlässigkeit und der Übertragungsreichweite berücksichtigt werden. Ein dritter Weg besteht darin, vorhandene Medien zu nutzen. Unter Powerline-Kommunikation (engl. Power Line Communication, PLC) wird die Kommunikation über die Niederspannungsleitungen des Hauses oder des Ortsnetzes verstanden. Möglich ist auch eine Vereinheitlichung durch IP-basierte Kommunikationswege (über LAN und WLAN).

### 2.3.2 Ansätze für Kommunikationsstrukturen im Internet der Dinge

Mit dem *Internet der Dinge* (engl. Internet of Things, IoT) sollen bisher nicht vernetzte Gegenstände und Geräte an das Internet angeschlossen werden (siehe Kapitel 1). Es gibt jedoch keine festen Vorgaben, was ein IoT-basiertes System umfassen soll. Die Quelle [MBR15] hat verschiedene Definitionen und typische Strukturen zusammenzutragen und kombiniert. Demnach bestehen IoT-Systeme, wie der Name suggeriert, zunächst aus Gegenständen



(engl. Things). Bei diesen kann es sich grundsätzlich um beliebige Arten von physikalischen Objekten handeln. Um ihre Eigenschaften und Funktionen zu teilen, sollten sie eindeutig identifizierbar sein und auf irgendeine Art, irgendwann und irgendwo (aber nicht immer und überall) mit anderen Instanzen verbunden sein. (vgl. [MBR15, S. 72-74])

Die Quelle [Ray18] beleuchtet u. a. verschiedene Anwendungsbereiche und typische Lösungen. Die aufgeführten Anwendungsfelder sind dabei weit gestreut: von der Warenverfolgung mit Funk-Identifikations-Chips (engl. Radio-Frequency Identification, RFID), der Analyse von landwirtschaftlichen Produktionen, der Überwachung des Straßenverkehrs mit Hilfe von Funk-Sensor-Netzen (engl. Wireless Sensor Networks, WSN) bis in das Gesundheitswesen (vgl. [Ray18, S. 9ff]).

Nach [WSJ15, S. 32] gibt es zwei grundsätzliche Ansätze für die Kommunikationsstruktur des IoT: Eine *zentralisierte Kommunikation* und eine *direkte Kommunikation*. Die zentralisierte Kommunikation wird auch *Client-Server-Kommunikation* genannt. Die direkte Kommunikation wird als *Peer-to-Peer-Kommunikation* (P2P) bezeichnet. Neben den zwei Ansätzen führt die Quelle noch hybride Ansätze auf, die beide Ansätze vermischen. (vgl. [WSJ15, S. 32-33], s. auch [MBR15, S. 59-60])

Bei der **direkten Kommunikation** zwischen zwei Geräten kommunizieren zwei gleichrangige Kommunikationspartner direkt und ohne zusätzliche Vermittler miteinander. In einigen Definitionen werden sogenannte Bridges als Sprach- bzw. Mediumsübersetzer geduldet. Wichtig ist, dass es keine zentralen Instanzen gibt, die die Kommunikation steuern oder überwachen. So schickt beispielsweise ein Lichtschalter nach Betätigung direkt eine Nachricht an eine (zuvor konfigurierte) Lampe, damit diese den Raum erleuchtet. Offensichtlich ermöglicht diese Variante eine minimale Latenz zwischen den Kommunikationspartnern. Dabei ist es erforderlich, dass die beiden Partner die gleichen Protokolle und Datenformate verwenden. (vgl. [MS06, S. 7], [BZ16]) Festzustellen ist, dass viele Literaturquellen es als unmöglich ansehen, Interoperabilität zwischen allen Geräten herzustellen. (vgl. [WSJ15, S. 33]).

Ein weiterer Vorteil der direkten Kommunikation besteht darin, dass Entfernungen und Nachbarschaften bestimmt werden können. So kann durch die Begrenzung des lokalen Netzsegments darauf geschlossen werden, welche Geräte sich in direkter Nachbarschaft befinden. Bei funkbasierten Protokollen kann über die Funklaufzeit sogar die Entfernung gemessen werden. (vgl. [WSJ15, S. 33])

Bei der zentralisierten Kommunikation verständigen sich zwei Aktionspartner über einen dritten zentralen Punkt miteinander. Beim HEMS (siehe Unterabschnitt 2.2.1) ist üblicherweise

ein lokales zentrales Gerät vorgesehen, dass diese Aufgabe erfüllt und das Energiemanagement durchführt. Im Internet der Dinge sollen weltweit alle Geräte miteinander in Aktion treten können. Deshalb haben sich Cloud-basierte Lösungen für die **zentralisierte Kommunikation** und Verarbeitung der Daten von angebundenen Gegenständen und Geräten etabliert. Der Begriff *Cloud* (engl. für Wolke) steht dabei für die Annahme, dass der Aufbau bzw. das Innere von Serverdiensten für die Nutzerinnen und Nutzer unbedeutend ist. Dies ermöglicht den Diensteanbietern Freiheiten, ihre Dienste dynamisch in Abhängigkeit von den aktuell benötigten Ressourcen zu betreiben. Ressourcen können inzwischen in feinen Granularitäten von Cloud-Providern angemietet werden. (vgl. [MBR15, S. 55-56]) Cloud-basierte Lösungen sind bei sozialen Netzwerken, E-Mail-Anbietern und inzwischen auch in IT-Systemen von Firmen stark verbreitet. Bei vielen Lösungen spricht man genauer vom (*mobilen*) *Cloud-Computing*. Dabei werden Ressourcen von (mobilen) vernetzten Geräten in die Cloud verlagert. Die zentrale Cloud übernimmt beispielsweise Rechen- oder Datenspeicherungsaufgaben für die Geräte, die als *Clouddienste* bereitgestellt werden. Diese Auslagerung hat den Vorteil, dass Geräte mit weniger Ressourcen ausgestattet werden können oder im Falle von mobilen Geräten durch die Auslagerung von Rechenaufgaben über geringere Akkukapazitäten verfügen können. (vgl. [Din+13, S. 1587-1589], für eine weitergehende Definition siehe [MG11])

Weitere Vorteile für Cloud-Computing sind u. a. die einfache Software-Bereitstellung der Funktionalitäten in Form von Clouddiensten aufgrund der zentralen Konzentration der Systeme, das automatische Backup aller zentral gesammelten Daten sowie die Durchsetzung von physischen Zugangskontrollen in Rechenzentren (vgl. [WSJ15, S. 32]). Aufgrund der zentralen Sammlung an Daten sind sogenannte *Big-Data-Analysen* einfach durchzuführen. Bei diesen Analysen geht es darum, mit Hilfe automatischer Lernverfahren aus einer großen Menge an Daten neue Erkenntnisse zu gewinnen oder sogar völlig neue Anwendungsfälle zu entwickeln. (vgl. [KTH13], [Jen17, S. 71], s. auch [Zha18])

Beeinträchtigungen gibt es hinsichtlich der zusätzlichen Verzögerung der Kommunikationsverbindung (Latenz). Selbstverständlich ist die komplette Struktur abhängig von der Zuverlässigkeit einer geeigneten Internetanbindung. Viele Literaturstellen weisen zudem auf Problematiken bei Sicherheit und Privatsphäre hin, wenn Daten von lokalen Geräten auf zentrale Geräte anderer Eigentümer übertragen werden. (vgl. [WSJ15, S. 32 und 34]) Weitere Quellen weisen auf die Abhängigkeit zum Diensteanbieter hin. Diese Abhängigkeit wird auch als Lock-in bezeichnet (vgl. [Sin10, S. 2]). Für eine weitere Abwägung der Vor- und Nachteile siehe auch [JG11].

Bei einer zentralisierten Kommunikation im Internet der Dinge übernimmt die Cloud die Funktion des Vermittlers zwischen den Funktionalitäten verschiedener Geräte. Durch die

Vorteile der zentralen Softwarebereitstellung können neue Vermittlungsfunktionen (Dienste) schnell zur Verfügung gestellt werden. Im Unterschied zur direkten Kommunikation müssen die Geräte nicht die gleichen Protokolle und Datentypen verarbeiten können. Die Vermittlungsfähigkeit und die schnelle Erweiterbarkeit erbringen große Vorteile hinsichtlich der Interoperabilität zwischen den Geräten. Nachbarschaften und Entfernungen zwischen Geräten sind innerhalb der Cloud jedoch nicht zuordenbar und müssen in der Regel erst konfiguriert werden (vgl. [WSJ15, S. 33]).

Neben diesen beiden Kommunikationsansätzen wird an **hybriden Ansätzen** geforscht. Beim (*mobilen*) *Edge-Computing* handelt es sich beispielsweise um einen Ansatz zur Erweiterung des (mobilen) Cloud-Computing. Dabei sollen die ausgelagerten Ressourcen durch zusätzliche, verteilte Rechneranmietungen näher am Endgerät bereitgestellt werden. Mit Edge (engl. für Kante) ist dabei der Rand der Kommunikationsinfrastruktur gemeint. Beim mobilen Edge-Computing soll dieser Bereich sogar bis in das Funkzugangsnetz (engl. Radio Access Network, RAN) hinein reichen. (vgl. [Abb+18, S. 450]) Vorteile gegenüber dem Cloud-Computing sind geringere Verzögerungen aufgrund der geringeren Distanz zwischen den Kommunikationspartnern. Zusätzlich werden weniger Ressourcen für die Datenübertragung bis zum Cloud-Dienstleister benötigt. Durch die verteilte Struktur beim Edge-Computing ergibt sich insbesondere für das IoT eine höhere Skalierungsfähigkeit sowie eine höhere Zuverlässigkeit im Vergleich zum Cloud-Computing. (vgl. [Abb+18, S. 453-454]) [Abb+18, S. 455 und S. 456] benennt neben dem IoT auch das Smart Grid und die intelligente Gebäudesteuerung (engl. Smart Building Control) als Anwendungsfelder für Edge-Computing.

*Fog-Computing* (engl. Fog für Nebel) wird in der Literatur teilweise synonym zum Edge-Computing verwendet. In einigen Veröffentlichungen ist die Definition weiter gefasst und umfasst auch Ressourcen beim Endnutzer zum Beispiel auf lokalen Netzwerkgeräten wie Routern oder Switchen. (vgl. [Has+18, S. 111])

Das Fog- und das Edge-Computing basieren auf der Konzeption des Cloud-Computing. So werden Anwendungen aus einer zentralen Sicht konzipiert und folgend in Programmteile aufgeteilt. Dabei wird davon ausgegangen, dass ein Programmteil auf jedem Rechenknoten (engl. Computing-Nodes) ausgeführt werden kann. Am besten wird ein Programmteil dabei auf dem Rechenknoten ausgeführt, der die geringsten Einbußen in der Latenz aufweist. (vgl. [Che+18, S. 96]) Die Forschung konzentriert sich auf die Netzwerkbereitstellung und auf die dynamische lastabhängige (Um-)Verteilung von Programmteilen (siehe z. B. [Bi+18] und [Che+18]).

Bisherige IoT-Geräte begnügten sich bislang mit dem Sammeln und Versenden von Daten. Durch Fortschritte in der Prozessortechnik kommen mehr und mehr Geräte auf den Markt,

die vollwertige Betriebssysteme ausführen können. Sie sind damit in der Lage erheblich mehr Aufgaben zu übernehmen und werden entsprechend als *intelligente Geräte* oder *intelligente IoT-Geräte* (engl. Smart IoT Devices) bezeichnet. (vgl. [Has+18, S. 110], [WSJ15, S. 32])

Abschließend sei angemerkt, dass beispielsweise [Jen17] einen gemeinschaftlichen neuen Ansatz für das IoT fordert. Dazu werden vier Grundbausteine bzw. Forderungen vorgeschlagen:

1. Intelligentes Einrichten: Das Hinzufügen eines neuen Gerätes oder auch einer größeren Anzahl an neuen Geräten muss für den Nutzer bzw. die Nutzerin beherrschbar bleiben. Oft kommen Techniken zum Einsatz, die die Nutzerin bzw. den Nutzer dazu auffordern Zugangspassworte abzutippen oder QR-Codes abzuscannen. Die Quelle schlägt vor, neuen Geräten direkt einen nicht-authentifizierten Zugang zum System zu ermöglichen, diesen jedoch vollständig einzuschränken bis dessen physische Existenz bestätigt ist. Abhängig vom verlangten Sicherheitsgrad könnte diese Bestätigung automatisch entfallen, wenn beispielsweise durch die Nutzerin bzw. den Nutzer eine mechanische Bestätigung erfolgt. Auch eine Funk-Entfernungsmessung könnte für eine Bestätigung verwendet werden.
2. Semantische Karte: Hier wird der Ansatz verfolgt, durch örtliche Nähe automatisch initiale Verknüpfungen oder zu mindestens Orientierungshilfen zu schaffen. So könnte beispielsweise eine Lampe automatisch mit dem nächstgelegenen Lichtschalter interagieren. Natürlich könnten solche Verknüpfungen durch die Nutzerin oder den Nutzer korrigiert werden.
3. Standardisierte Beschreibungen: Geräte sollen ihre Fähigkeiten und Eigenschaften dem restlichen System bekannt geben, sodass die Systemkomponenten erkennen von welchem Nutzen das Gerät für sie sein kann. Standardisierte Beschreibungen könnten dabei helfen, ein einheitliches Verständnis von Fähigkeiten und Eigenschaften zu ermöglichen.
4. Zentralisierte Daten: Daten, besonders Sensordaten, können für viele verschiedene Anwendungen hilfreich sein. Hierzu gehören nicht nur vorübergehende Events, sondern auch vergangene Daten z.B. für das maschinelle Lernen. Ein zentraler Datenpool könnte hier eine Lösung sein. Allerdings wird hier der Datenschutz als entscheidende Problematik benannt.

### 2.3.3 Middleware für die Anbindung verteilter Geräte

Damit bei einer zentralisierten Kommunikation (lokal oder Cloud-basiert) verschiedene verteilte Geräte mit unterschiedlichen Protokollen und Datenstrukturen miteinander in Aktion treten können, bedarf es einer Übersetzungsschicht. Solche Übersetzungsschichten können sowohl innerhalb einer Implementierung, aber auch als eigene Netzwerkdienste auftreten. Eine solche Schicht wird im englischen als *Middleware* bezeichnet. Sie verschleiert für die höheren Anwendungsschichten den Zugriff auf die einzelnen verteilten Subsysteme. (vgl. [Raz+16, S. 70]) Die Theorie der Middleware entstammt dem Themengebiet der verteilten Systeme (vgl. [ST17, S. 5]).

Die Quelle [Raz+16, S. 76] unterscheidet sieben grundlegende Lösungsansätze für IoT-Middleware-Schichten. Auf drei Ansätze soll hier weiter eingegangen werden: *Ereignis-basiert*, *Dienst-orientiert* und *VM-basiert*. Eine Plattform kann dabei auch auf mehreren Lösungsansätzen basieren. Nach der Quelle zeigen Plattformen aus mehreren Lösungsansätzen üblicherweise eine bessere Leistung als Plattformen aus nur einem Lösungsansatz, da sie die Vorteile verschiedener Ansätze vereinigen.

Bei *Ereignis-basierten Plattformen* kommunizieren die Komponenten durch sogenannte *Events* (engl. für Ereignis) miteinander. Ein Event enthält Informationen über eine Statusänderung seines Erzeugers. Dabei werden der Typ des Events und weitere beschreibende Daten übergeben. Ein Event kann zu mehreren Empfängern weitergeleitet werden. Da es keinen festen Empfänger gibt, ist eine direkte Antwort nicht möglich. Hier wird oft das Publish-Subscribe-Entwurfsmuster (vgl. [Eug+03]) für die Handhabung und Datenübermittlung verwendet. Dabei abonniert eine Komponente eine Event-Quelle und erhält fortan automatisch die dort veröffentlichten Events. (vgl. [Raz+16, S. 76], [MBR15, S. 60-62], siehe auch [ST17, S. 66-71])

*Dienst-orientierte Plattformen* basieren auf der Dienst-orientierten Architektur (engl. Service-Oriented Architecture, SOA), welche eine hohe Verbreitung in den IT-Systemen von Firmen gefunden hat. Dabei bieten die Komponenten Dienste im (lokalen) Netzwerk an, die von anderen Komponenten angesprochen werden können. Durch die Koordinierung und Verknüpfung dieser Dienste können dann Funktionalitäten ermöglicht werden. (vgl. [Raz+16, S. 81], siehe auch [ST17, S. 62-66])

VM-basierte Plattformen erlauben die sichere Ausführung von Anwendungen als virtuelle Maschinen (VM). Dazu werden die Anwendungen in Anwendungsteile unterteilt und diese verteilt auf den Geräten im Netzwerk ausgeführt. Dabei erfolgt die Virtualisierung entweder zwischen dem Betriebssystem und den Anwendungen oder ersetzt sogar vollständig das Betriebssystem.

Konkrete Systeme zu den verschiedenen Middleware-Ansätzen sowie deren Vor- und Nachteile in Hinsicht auf das IoT sind in der Quelle [Raz+16] aufgezählt.

### 2.3.4 Protokolle für das Internet der Dinge

Nachfolgend werden offene bzw. standardisierte Protokolle vorgestellt, die im Kontext des Internet der Dinge häufig benannt werden.

*Schnittstellen zum Aufruf entfernter Funktionen* (engl. Remote Procedure Call, RPC) sind der übliche Weg Daten zwischen Komponenten in einer **Dienst-orientierten Architektur** auszutauschen. Dabei ruft eine Komponente direkt die Schnittstelle einer anderen Komponente auf und erhält darauf das Resultat der entfernten Funktion als Antwort. Andere Ausprägungen sind unter anderem *Representational State Transfer* (engl., REST) und das *Simple Object Access Protocol* (engl., SOAP). (vgl. [MBR15, Seiten 62–64])

SOAP baut auf einer Client-Server-Kommunikation auf. Hier wird üblicherweise das Hyper Text Transfer Protocol (engl., HTTP) verwendet, es kommen aber auch andere Protokolle zum Einsatz. Für die Anfrage und die Übermittlung von Objekten verwendet SOAP ausschließlich die Extensible Markup Language (engl., XML). Die Spezifikation von SOAP beinhaltet auch die Web Services Description Language (engl., WSDL). Mit Hilfe dieser Sprache kann eine SOAP-Schnittstelle beschrieben und automatisch eingebunden werden. Im Gegensatz zu vielen anderen Protokollen enthält SOAP definierte Strukturen zur Fehlerhandhabung und -übermittlung. (vgl. [MBR15, S. 63])

Das Übertragungsprotokoll REST verwendet ebenfalls HTTP als Basis. Im Gegensatz zu SOAP werden dabei die Ressourcen nicht über eine XML-Anfrage, sondern direkt über eine URL adressiert. REST verwendet die vier HTTP1.1-Anfragetypen GET, POST, PUT und DELETE zum Lesen, Ändern, Erstellen und Löschen von Objekten. REST spezifiziert nicht, in welchen Datenformaten die Objekte formatiert sind. Neben XML, sind JavaScript Object Notation (engl, JSON), Really Simple Syndication (engl, RSS) und viele weitere zu finden. (vgl. [MBR15, S. 63-64])

Das *Message Queuing Telemetry Transport* (engl., MQTT) Protokoll implementiert das Publish-Subscribe-Entwurfsmuster (vgl. [BG14, S. 1-2]) der **Ereignis-basierten Architektur**. Es wurde als offenes und einfach zu implementierendes Protokoll zur Übermittlung von Sensordaten mit wenig Overhead für das IoT entwickelt. Das Protokoll setzt auf TCP/IP auf; kann aber auch über andere stream-basierte bidirektionale Netzwerkprotokolle übertragen werden. Bei MQTT werden Nachrichten auf sogenannte Topics (engl. für Themen) veröffentlicht. Ein Abonnent kann spezifische Topics oder mittels Platzhalter gleichzeitig mehrere

Topics abonnieren. Bei der Nachrichtenübermittlung unterstützt MQTT drei verschiedene Dienstgüten (engl. Quality of Service, QoS), die nach der folgenden Reihenfolge jeweils mehr Netzwerkverkehr erfordern: (vgl. [BG14, S. 1-2], [MBR15, S. 65-68])

- „Höchstens einmal“ (QoS = 0): Es erfolgt ein Vermittlungsversuch. Paketverluste werden nicht abgefangen. Der Übermittlungserfolg ist abhängig von der Qualität der zugrundeliegenden Infrastrukturen.
- „Mindestens einmal“ (QoS = 1). Bei der Übermittlung wird sichergestellt, dass die Daten beim Zielsystem ankommen. Es können doppelte Übermittlungen auftreten.
- „Genau einmal“ (QoS = 2). Hier wird zusätzlich sichergestellt, dass doppelte Nachrichten nicht an Abonnenten weitergeleitet werden.

Zusätzlich unterstützt MQTT weitere Merkmale, wie Verschlüsselung, Authentifizierung und Funktionalitäten zur Verbindungsüberwachung.

Das *Constrained Application Protocol* (engl., CoAP) ist angelehnt an REST. Es identifiziert Objekte ebenfalls anhand einer URL und nutzt die vier oben benannten HTTP-Anfragetypen. Anders als REST bzw. HTTP setzt es jedoch auf einem kompakten Binärformat für die Übertragung auf. Das komplette Protokoll wurde für einen geringen Ressourcenverbrauch optimiert. So setzt es auf UDP auf und unterstützt optional ähnliche Dienstgüten wie MQTT. CoAP wurde anders als REST für die **direkte Kommunikation** konzipiert. Der verfolgte Ansatz dabei ist, dass ein Sensor eine Entscheidung trifft (zum Beispiel, dass ein Mensch anwesend ist) und mit CoAP einen Befehl an einen oder mehrere Aktoren sendet (um zum Beispiel das Licht einzuschalten). (vgl. [MBR15, S. 68])

Weiterhin finden viele weitere Protokolle im IoT Anwendung.

- Beispielsweise *Extensible Messaging and Presence Protocol* (engl., XMPP), ein Protokoll entwickelt für Chat-Clients (siehe z. B. [Con+12, S. 3]),
- *Advanced Message Queue Protocol* (engl., AMQP), ein Protokoll entwickelt für den Finanzsektor,
- *IPv6 over Low power Wireless Personal Area Network* (engl., 6LoWPAN) ein IPv6-basiertes Protokoll für Wireless Sensor Networks sowie
- *Universal Plug and Play* (engl., UPnP) (siehe z. B. [KKA19, PP S. 2]), ein Protokoll zur automatischen Lokalisierung, Adressierung und Steuerung von Datenobjekten in einem lokalen Netzwerk. (vgl. [Eja+17, S. 86], [Ray18, S. 25]).

### 2.3.5 IoT- und Cloud-basierte Energiemanagementlösungen

In der Literatur finden sich bereits verschiedene Ansätze, automatisierte Energiemanagementsysteme (HEMS, siehe Unterabschnitt 2.2.1) auf Cloud-basierte oder IoT-basierte Strukturen aufzusetzen. Die veröffentlichten Lösungen besitzen viele unterschiedliche Ziele, Lösungswege und Implementierungen. Viele der Literaturquellen identifizieren Systemkomponenten oder -dienste. Da sich die Lösungsansätze und die Fokussierung der Quellen deutlich unterscheiden, beschreiben die Quellen verschiedene Komponenten bzw. Dienste, deren Vergleich deshalb nicht zielführend ist. Nachfolgend werden deshalb zunächst verschiedene Systemtypen unterschiedenen, um zumindest die verwendeten Einsatzmöglichkeiten des Internet der Dinge zu verdeutlichen. Die Aufteilung ist an die Quelle [KSG18] angelehnt:

- **Systemtyp 1** sieht einen lokalen Server vor, der optional Rechenressourcen in die Cloud verlegen kann. Dadurch können Funktionalitäten der Cloud, wie Echtzeitpreise und Direkteingriffe genutzt werden. (vgl. [KSG18, Szenario 1]) Der lokale Server stellt damit eine zentralisierte Komponente dar. Wegen seiner dezentralen Lokalisierung und der Unterteilung der Kommunikation auf mehrere Ebenen werden Systeme des Typs in dieser Arbeit zu den hybriden Kommunikationsansätzen (siehe Unterabschnitt 2.3.2) gezählt.

Ähnliche Implementierungen finden sich z. B. in [AB18], [KKA19]

- **Systemtyp 2** sieht einen Cloud-Dienst und ein HEMS-IoT-Gateway vor. Das Gateway leitet *alle* Informationen (von Messgeräten) ins Internet an den Cloud-Dienst weiter. Dieser führt die Berechnungen der Optimierung durch und leitet Kontrollbefehle zur Steuerung der Geräte an das Gateway zurück. (vgl. [KSG18, Szenario 2]) Dieser Typ setzt auf einer zentralisierten Kommunikationsstruktur (siehe Unterabschnitt 2.3.2) auf.

Ähnliche Implementierungen finden sich z. B. in [Sou+18], [Xia+18], [LL16], [PLT16].

- **Systemtyp 0** (in dieser Arbeit): HEMS-Lösungen nach einem reinen lokalen Entwurf (wie in Unterabschnitt 2.2.1 vorgestellt), die jedoch durchaus Ideen oder Protokolle des IoT anwenden.

Um den Stand der Forschung weiter zu verdeutlichen, wird nachfolgend eine kleine Auswahl an konkreten Lösungen aus der Literatur vorgestellt:

[KKA19] beschreibt ein Gateway nach Systemtyp 1 für das Energiemanagement. Im Heimnetzwerk wird das Universal Plug & Play (engl., UPnP) Protokoll zur Kommunikation mit intelligenten Geräten verwendet. Dies ermöglicht eine automatische Zuordnung und Konfiguration (engl. Auto Discovery und Zero Configuration) der intelligenten Geräte. Für die



Kommunikation mit den Cloud-Diensten (des Netzbetreibers) wird ein mit einer elliptischen Kurvenverschlüsselung abgesichertes SOAP über HTTP verwendet. Das Gateway realisiert dabei ein lokales Energiemanagement. Die Kontrolle einiger Geräte kann hierzu an die Cloud-Dienste abgegeben werden. Die Cloud-Dienste übernehmen zusätzlich die Bereitstellung von Echtzeit-Energiepreisen.

[AB18, S. 19] führt eine Struktur aus zentraler und lokaler Verarbeitung nach Systemtyp 1 ein. Für die Kommunikation zwischen Geräten und dem zusätzlich eingeführten Smart-Gateway (für die lokale Verarbeitung) wird Bluetooth verwendet. Die Kommunikation zwischen lokaler und zentraler Verarbeitung verwendet das HTTP-Protokoll. In der Testumgebung wird die zentrale Verarbeitung bei einem realen Cloud-Anbieter ausgeführt.

[Xia+18] wendet nach Systemtyp 2 die Idee des Edge-Computing auf das Energiemanagement von Privathaushalten an. Dabei werden Dienste für das Energiemanagement identifiziert und beschrieben, wie z. B. eine Energieerzeugungsvorhersage für Solaranlagen und ein Klassifizierungsmodul zur Einordnung von Geräten. Das Energiemanagement profitiert dadurch, da durch das Edge-Computing auf Wetterinformationen zurückgreifen kann und die Möglichkeit zur Verwendung verschiedener Geräte und Techniken besteht. Anhand einer Testumgebung wird das implementierte System analysiert.

[Sou+18] beschreibt ein Konzept nach Systemtyp 2 für das Energiemanagement in Privathaushalten. Dabei wird das Protokoll MQTT zur Kommunikation mit intelligenten Geräten verwendet. Die Geräte besitzen WLAN-Funkmodule. Das Konzept sieht getrennte Controller für das Management von Energieerzeugern und -verbrauchern vor. Das Energiemanagement wird auf der Basis einer Fuzzy-Logiksteuerung umgesetzt. Zusätzlich wird ein Nutzerinterface vorgestellt.

Quellen wie [AV16] und [Xia+18] verwenden Prinzipien des Edge- bzw. des Fog-Computing (siehe Unterabschnitt 2.3.2) bei der Konzeption von Energiemanagementsystemen. Bei der Verwendung dieser Prinzipien vermischen sich zunehmend die Systemtypen 1 und 2. Deutlich wird dies z. B. bei der Lösung in [AV16]. Dort werden Energiemanagement-Funktionen auf Basis einer Fog-Computing-Architektur unter der Verwendung einer Ereignis-basierten Middleware implementiert. Die Funktionen des Energiemanagements werden als Dienste implementiert und können auf lokalen oder entfernten Routern ausgeführt werden. Durch die Verwendung von offenen Web-Standards kann die Interoperabilität erhöht werden. Die Fog-Architektur ermöglicht das Einhalten von Echtzeit-Anforderungen und erhöht die Skalierbarkeit des Gesamtsystems. (vgl. [AV16, S. 168])

[CD17] wendet Edge-Computing und (virtuelle) IoT-Geräte auf das Anwendungsfeld der Hausautomation an. Ein Energiemanagement ist hier nicht Teil der Implementierung. Der

Entwurf folgt dem Systemtyp 2. Für die Edge-Computing-Plattform werden Komponenten identifiziert und beschrieben.

Aus der Unterschiedlichkeit der beschriebenen Lösungen wird die Schwierigkeit deutlich, eine Interoperabilität sowohl zu Geräten, aber auch zwischen Systemen herzustellen. Jede Lösung verwendet andere Kommunikationstechniken, um mit Geräten zu kommunizieren. Bei vielen dieser betrachteten Lösungen werden offene Protokolle verwendet, wie MQTT (siehe z. B. [Sou+18]), CoAP (siehe z. B. [LL16]), REST (siehe z. B. [KSG18]) und SOAP (siehe z. B. [KKA19]), was auch eine Interoperabilität zwischen Systemen ermöglichen könnte. Dennoch weisen die Beschreibungen darauf hin, dass die Schnittstellen zwischen den Komponenten jeweils spezifisch auf ihre Anwendungsfälle hin zugeschnitten wurden. Neue Anwendungsfälle bzw. Dienste bedürfen wahrscheinlich einer Erweiterung der Schnittstellen.

Sicherheits- und Datenschutzanforderungen der lokalen Nutzerinnen und Nutzer werden in den benannten Quellen bisher kaum beachten. Im Bereich der Energie-Informationssysteme (siehe Unterabschnitt 2.2.3) gibt es einige Ansätze Sicherheits- und Datenschutzanforderungen zu gewährleisten. Die Quelle [Lee+17] schlägt z. B. eine Kommunikationsstruktur vor, die als Gemeinschaftshierarchie (engl. Community Hierarchy) bezeichnet wird. Mehrere Haushalte werden dabei abhängig von ihrem Ort mit einem Gemeinschafts-Broker verbunden. Diese Broker sind wiederum mit einer Cloud-Plattform verbunden. Auf allen Ebenen sorgen Zugriffsberechtigungen und Datenaggregationen dafür, dass keine sensiblen Daten in der Hierarchie weiter nach oben transportiert werden. Ähnliche Ansätze und Protokolle werden in [LLL11] und [Wan+18] vorgestellt.

Die Lösung aus [Lee+17] erlaubt Daten über eine Gemeinschaft zusammenzustellen, ohne dass Rückschlüsse auf individuelle Haushalte möglich sind. Die Plattform wurde spezifisch für die Datenanalyse entwickelt. Steuerungsfunktionen sind nicht vorgesehen. (vgl. [Lee+17]) Da die zum Energiemanagement notwendigen individuellen Daten und Funktionen der Haushalte nicht in der Cloud-Plattform vorliegen, kann dort kein Energiemanagement nachgerüstet werden. Es müssten alle Plattformen und Schnittstellen erweitert werden, um die verschiedenen Energiemanagement-Funktionalitäten hinzuzufügen.

## 3 Ansätze zur Erweiterung von Energiemanagementsystemen

Damit die Forderungen der Bundesregierung bzw. der EU zur Umsetzung der Energiewende (siehe Kapitel 1 und Unterabschnitt 2.1.1) erreicht werden können, ist eine möglichst großflächige Verbreitung von Energiemanagementsystemen anzustreben. Das anvisierte Ziel ist, mit Energiemanagementsystemen die Energieverbräuche in möglichst vielen Umgebungen möglichst weitgehend zu optimieren. Dieses Ziel wird als eine mögliche Vision in Abschnitt 3.1 weiter ausgearbeitet. Die Vision basiert auf Argumentationen aus Kapitel 2 sowie weiteren Literaturstellen.

Wie im vorherigen Kapitel dargestellt wurde, gibt es in der Literatur bereits eine große Anzahl an Konzepten. Zunehmend sind auch Produkte zum Energiemanagement auf dem Markt verfügbar. Diese Energiemanagementsysteme stellen für bestimmte Anwendungsfälle bereits funktionierende Lösungen dar. (siehe Abschnitt 2.2 und 2.3)

Jedoch weisen die Systeme vielfach strukturelle Defizite auf, welche in Abschnitt 3.2 im Vergleich zur beschriebenen Vision herausgestellt und analysiert werden. Demnach unterliegen bestehende Energiemanagementsysteme Beschränkungen hinsichtlich der erreichbaren Optimierung sowie der Umgebungen, in denen sie eingesetzt werden können. Auch sind Forderungen zur Einhaltung des Datenschutzes bzw. der Datensicherheit zu berücksichtigen. Diese Defizite können zu Einschränkungen in ihrer Einsatzfähigkeit führen, die im ungünstigsten Fall sogar der Anschaffung solcher Systeme im Weg stehen. Zusätzlich werden in diesem Kontext die in Unterabschnitt 2.3.5 beschriebenen Systemtypen auf ihre Tauglichkeit hin analysiert.

Wegen der fehlenden großflächigen Einsatzfähigkeit bisheriger Lösungen ist eine grundsätzlichere Betrachtung des Themas Energiemanagement auf Basis einer verteilten Gerätevernetzung sinnvoll. Wie bereits in Kapitel 1 dargelegt, wird zur effizienten Kombination möglichst vieler Optimierungsansätze ein universelles und flexibles Systemkonzept benötigt. Dieses Zielsystem für zukünftige Energiemanagementsysteme wird in Abschnitt 3.3 näher erläutert.

Für die Verwirklichung des Systemkonzepts und zur späteren Überprüfung der Zielerreichung werden von Unterabschnitt 3.3.1 bis 3.3.4 Ziele und Anforderungen formuliert.

### 3.1 Eine Vision für Energiemanagementsysteme

Für die Realisierung eines möglichst „idealen“ Energiemanagements ist eine effiziente und zielgerichtete Kommunikation zwischen den Menschen, den verwendeten Geräten und den Energieerzeugern anzustreben. Der Mensch profitiert dabei von einem möglichst hohen Komfort und minimalen Energiekosten. Ihre Funktion stellen die Geräte in Abhängigkeit von anderen Geräten und Daten von Energieerzeugern automatisch so ein, dass langfristig möglichst wenig Energie erzeugt werden muss. Der Mensch erhält dabei immer einen Überblick über die Aktivitäten seiner Geräte. Er ist zudem informiert über den Energieverbrauch der Geräte und kann z. B. daraus folgern, welche Veränderungen seiner Nutzungsmuster sinnvoll sind oder welche Geräte gegen effizientere Komponenten ausgetauscht werden könnten.

Dazu sollten folgende Punkte erfüllt sein:

- Die Etablierung eines Energiemanagements sollte **in allen Umgebungen** möglich sein, in denen Energieverbraucher zu finden sind.<sup>1</sup> Darunter sind Privathaushalte zu benennen, wie Ein- und Mehrfamilienhäuser sowie Miet-, Zweit- und Ferienwohnungen. Weiterhin kommen auch kleine und mittelständische Unternehmen sowie Konzerne verschiedener Branchen mit und auch ohne eigene Energiemanagerinnen und -manager in Frage. Zusätzlich sollten auch öffentliche Einrichtungen und andere Gemeinschaftsumgebungen beachtet werden.
- Zukünftige Energiemanagementsysteme sollten in der Lage sein, den verschiedenen Stakeholdern zu helfen, ihre aus den Beweggründen entwickelten Ziele zu realisieren (siehe Unterabschnitt 2.1.2). Ein weiteres Ziel ist außerdem, durch die Kombination mehrerer Ansätze ein erweitertes Optimierungspotential ausschöpfen zu können. Aus diesen zwei Gründen sollten **verschiedenste Optimierungsansätze** sowohl für das automatisierte Energiemanagement als auch zur Nutzer-Information ermöglicht werden.<sup>2</sup>
- Damit ein möglichst hohes Optimierungspotential in einer Umgebung gehoben werden kann, sollte ein Energiemanagementsystem offensichtlich in der Lage sein, **möglichst viele Geräte einer Umgebung** einzubinden. Die Anbindung sollte dabei unabhängig

<sup>1</sup>Ein mehrfacher Einsatz ergibt Synergieeffekte, die bereits in der Einleitung zu Abschnitt 2.2 festgestellt wurden.

<sup>2</sup>Auch diese Synergieeffekte wurden in der Einleitung zu Abschnitt 2.2 benannt.

von Herstellern, Gerätearten oder bestimmten Kommunikationsschnittstellen erfolgen können. Neben neuartigen intelligenten Geräten sollten auch herkömmliche Geräte mit und ohne Kommunikationsschnittstellen integriert werden können.

- Des Weiteren sollte das Energiemanagement **über räumliche Grenzen hinweg mit vergleichbaren Systemen interagieren** können. Diese Kooperation wurde bereits in Unterabschnitt 2.1.2 gefordert. Sie ist z. B. bei großen und mittelständischen Unternehmen mit vielen, auch örtlich verteilten, Standorten und unterschiedlichen Umgebungen interessant. Aber auch im privaten Kontext wird diese Forderung wichtiger. Als Beispiel sei hier die Mieterstromförderung genannt, die eine Koordination zwischen einer gemeinschaftlichen Solaranlage und verschiedenen Mietparteien motiviert (siehe dazu Unterabschnitt 2.1.2). Eine solche Kooperation könnte zukünftig u. a. Mehrparteienhäuser aber auch Firmengelände, Straßenzüge und sogar Orts- oder Stadtteile überdecken.
- Sowohl die Errichtung wie auch der Betrieb und die Instandhaltung eines Energiemanagementsystems sollten möglichst geringe Kosten verursachen. Eine Möglichkeit wäre, das Energiemanagement als integralen Bestandteil anderer ebenfalls vorhandener Infrastrukturen zu konzipieren. Synergieeffekte wären auch durch eine universelle Infrastruktur für die Nutzung mehrerer Anwendungsfelder möglich. Diese Synergieeffekte wurden bereits im Kontext des Internets der Dinge in der Einleitung zu Abschnitt 2.3 diskutiert.
- Aufgrund der großen Anzahl der Geräte sollten Verknüpfungen zwischen den Geräten und den Funktionalitäten des Energiemanagements automatisch anhand der Geräte-Eigenschaften erfolgen. Diese automatische Verknüpfung wird als *Auto Discovery* (engl. für automatische Erkennung) bezeichnet. Damit wird das Ziel verfolgt, die Notwendigkeit von Eingriffen durch den Menschen soweit wie möglich zu minimieren. Diese Forderungen finden sich z. B. auch in [KKA19, S. 1] und [Jen17, S. 71].
- Damit sich ein solches Energiemanagementsystem stets an den Wünschen der lokalen Nutzerinnen und Nutzer ausrichtet, sollten diese die Kontrolle über das System in der lokalen Umgebung besitzen. Diese Forderung findet sich ebenfalls auch in [Jen17, S. 71]. Die Nutzerinnen und Nutzer könnten dabei selbst bestimmen, welche Funktionalitäten ihr Energiemanagement umfassen soll. Der Zugriff durch Dritte sollte durch die Nutzerinnen und Nutzer eingeschränkt werden können. Optimal wäre, wenn eine Zugriffserlaubnis durch Dritte erst durch die Nutzerinnen und Nutzern eingeräumt werden muss.
- Weiterhin ist ein möglichst hoher Schutz der Privatsphäre der Nutzerinnen und Nutzer

anzustreben, wie bereits in der Einleitung zu Abschnitt 2.3 hervorgehoben wurde. Aufgrund der hohen Vernetzung des Energiemanagements mit vielfältigen Geräten stehen hier besonders viele Daten und Funktionen zur Verfügung, die aufgrund ihrer Nähe zu den Nutzerinnen und Nutzern Aussagen über deren Handlungen und Gewohnheiten erlauben könnten. Ähnliches gilt auch für den Schutz von Betriebsgeheimnissen, die aus der Überwachung bzw. Steuerung von Fertigungsprozessen abgeleitet werden könnten. Auch auf einen hohen Schutz vor Fremdzugriff für Funktionen zur Steuerung von Geräten und Anlagen ist zu achten.

## 3.2 Analyse aus der Literatur bekannter und am Markt vorhandener Lösungsansätze

Im Vergleich zwischen der beschriebenen Vision (siehe Abschnitt 3.1) und dem Stand der Technik werden die folgenden strukturelle Defizite bezüglich der Einsatzfähigkeit bisheriger Energiemanagementsysteme deutlich. Dazu werden insbesondere die Erkenntnisse aus Abschnitt 2.2 und 2.3.5 zugrunde gelegt.

**Vorhandene Lösungen fokussieren jeweils nur auf Teilaspekte:**

- Viele Systeme implementieren nur spezifische Optimierungsansätze (siehe Unterabschnitt 2.2.1 und 2.2.2). So legt das HEMS (siehe Unterabschnitt 2.2.1) seinen Schwerpunkt beispielsweise auf den Optimierungsansatz der automatisierten Energieverbrauchsoptimierung und vernachlässigen Nutzer-unterstützende Funktionen.
- Teilweise werden die Interessenlagen bestimmter Stakeholder berücksichtigt – und andere wiederum vernachlässigt (siehe Unterabschnitt 2.1.2 und 2.1.3). So werden zum Beispiel beim HEMS die Anwenderinnen und Anwender der Geräte nur beiläufig berücksichtigt.
- Einige Systeme fokussieren auf bestimmte Umgebungen. Auch hier adressiert das HEMS mit der Eigenverbrauchsoptimierung in der Regel nur Einfamilienhäuser.
- Auch verwenden vorhandene Lösungen spezifische Techniken zur Einbindung von Geräten – und lassen wiederum andere Techniken außer Acht (siehe Unterabschnitt 2.2.1). Die Interoperabilität zwischen den Systemen und den einzubindenden Geräten ist also eingeschränkt.

Durch die Fokussierung auf Teilaspekte ergeben sich eine Vielzahl von Abhängigkeiten zwischen den möglichen Optimierungsansätzen, den Umgebungen und den Techniken zur

Einbindung von Geräten. Weiterhin sind die adressierten Stakeholder bei der Wahl eines geeigneten Energiemanagementsystems zu berücksichtigen.

**Viele Systeme auf dem Markt verstehen sich als geschlossene Gesamtlösungen** mit internen Schnittstellen und eingeschlossenen Daten, ebenso wie bisherige Lösungen mit IoT-Ansatz (siehe [WSJ15, S. 29]). Dadurch sind diese Systeme in der Regel nur durch die jeweiligen Anbieter erweiterbar. Wie in Unterabschnitt 2.3.1 benannt, hat u. a. diese Problematik im Bereich des Smart Home zu einer Zersplitterung des Marktes von Smart-Home-Produkten und dadurch zu einer Minderung der Verbrauchererfahrung geführt.

Zusammenfassend ergibt sich daraus die Problematik, dass sich bisherige Lösungen in vielen vorhandenen Umgebungen mutmaßlich nicht rentieren, sehr lange Amortisationszeiten besitzen, oder nur für den Einsatzzweck nur eingeschränkte Funktionalitäten bieten. Im ungünstigsten Fall könnte diese Problematik sogar der Anschaffung solcher Systeme im Weg stehen. So könnten die vorhandenen Kommunikationstechniken einer bereits vernetzten Umgebung eine große Einschränkung in der Wahl eines passenden Energiemanagementsystems darstellen. Gleiches gilt, wenn bestimmte Optimierungsansätze gewünscht sind oder bestimmte Stakeholder adressiert werden sollen. Im Zweifelsfall müsste der Austausch der gesamten vorhandenen Kommunikationstechnik und/oder der Geräte erwogen werden, um beispielsweise einen bestimmten Optimierungsansatz anwenden zu können.

Auch lassen sich mehrere gleichzeitige Optimierungsansätze oder Geräteeinbindungen bei simultanem Betrieb mehrerer Energiemanagementsysteme nur eingeschränkt erzielen. Zunächst erhöht der Betrieb mehrerer Systeme die Kosten bei Anschaffung und Betrieb. Die betrachteten Lösungen (in Abschnitt 2.2 und 2.3) weisen zudem keine Schnittstellen für eine Kooperation und damit auch keine Interoperabilität zwischen Energiemanagementsystemen auf. Auch die Systembeschreibungen derartiger Systeme sehen bisher keine Kooperation zwischen Systemen vor. Mit der Norm IEC 61970 existiert aber grundsätzlich ein Standard zum Datenaustausch von Energieverbrauchsdaten und Steuerbefehlen (siehe [Ber02]). Aus dem Grund der fehlenden Interoperabilität ist auch eine feinere Untergliederung z. B. in Betriebsstätten, Arbeits- und Büroräume oder auch einzelne Arbeitsplätze in der Regel nicht möglich. Eine solche Separation von Umgebungen wäre für eine bessere Skalierungsfähigkeit hilfreich.

Ein Lösungsansatz zur Überwindung der eingeschränkten Erweiterbarkeit wäre der Aufbau von Energiemanagementsystemen auf Basis einer offenen Plattform. Eine solche Plattform müsste es ermöglichen, dass verschiedene Anbieter Komponenten beitragen können. Einen solchen Ansatz verfolgt z. B. [Sow+18]. In der Quelle werden Komponenten zum Energiemanagement in ein interoperables quelloffenes Hausautomationstool eingefügt. Damit

werden eine Gerätesteuerung sowie eine Visualisierung des Energieverbrauchs ermöglicht. Das Energiemanagement über mehrere Umgebungen hinweg bleibt dabei aber unbeachtet.

Kann ein System Anforderungen an den Datenschutz bzw. die Datensicherheit nicht berücksichtigen (wie in der Vision begründet), kann auch dies einen Hinderungsgrund für die Anschaffung darstellen.

Wie in Unterabschnitt 2.3.5 beschrieben wurde, unterscheiden sich Energiemanagementsysteme in verschiedene Systemtypen. Die Systemtypen 1 und 2 binden nach dem Internet der Dinge zentralisierte Komponenten mit ein. Aufbauend auf den Kommunikationsstrukturen aus Unterabschnitt 2.3.2 sollen nachfolgend die Vor- und Nachteile der verschiedenen Systemtypen gegenübergestellt werden. Dabei sollen insbesondere die in der Einleitung zu Abschnitt 2.3 benannten Begriffe Interoperabilität und Datenschutz näher betrachtet werden.

- **Lokale zentralisierte Kommunikation (Systemtyp 0):** Ein lokales System zum Energiemanagement, wie in Unterabschnitt 2.2.1 vorgestellt, besitzt den Hauptvorteil, dass es in der Kontrolle der Eigentümerin bzw. des Eigentümers liegt. Aufgrund der lokalen Datenhaltung müssen in der Regel keine Daten an externe Dienstleister übermittelt werden, welches einen hohen Schutz der Privatsphäre bzw. von Geschäftsgeheimnissen ermöglicht. Wie bereits in der Einleitung zu Kapitel 2 erläutert wurde, ist die Interoperabilität bei der Anbindung von Geräten eine der größten Herausforderungen bei diesem Systemtyp. Auch die Kooperation mit anderen Umgebungen oder mit gemeinschaftlich genutzten Geräten außerhalb der Umgebung ist in der Regel bei diesem Systemtyp nicht vorgesehen (siehe Unterabschnitt 2.2.1).
- **Zentralisierte Kommunikation (Systemtyp 2):** Wie in Unterabschnitt 2.3.5 verdeutlicht wurde, kann mit den Basisideen des Internet der Dinge die Interoperabilität sowohl in Verbindung mit Geräten, als auch zwischen Energiemanagementsystemen in verschiedenen Umgebungen verbessert werden. Wie [AB18, S. 18] aufführt, treten dabei jedoch höhere Anforderungen nach Datenraten, Probleme durch variierende und höhere Kommunikationsverzögerungen sowie eine erhöhte Anfälligkeit für Cyberangriffe auf. Zusätzlich führt die Quelle die großen entstehenden Datenmengen an, wenn alle aufkommenden Daten an Cloud-Dienste versendet und dort gespeichert werden. Cloud-Dienste müssen entsprechend der zunehmenden Anzahl der IoT-Geräte mitwachsen (Skalierung). Durch die Übertragung aller Daten ins Internet bzw. an externe Dienstleister werden dabei womöglich sensible Daten versendet, wodurch Datenschutzanforderungen nicht eingehalten werden können (siehe Unterabschnitt 2.3.5 sowie [KKA19, S. 1] und [Err+18, S. 4381]).



- **Hybride Ansätze (Systemtyp 1, sowie Systemtyp 2 mit Edge-/Fog-Computing-Ansatz):** Bei diesen Systemtypen wird das Ziel verfolgt einigen Nachteilen der zentralisierten Kommunikation entgegenzuwirken. Aufgrund der verteilten Struktur skaliert die gesamte Struktur besser als bei Systemtyp 2. Solche Systeme sind im Unterschied zu zentralisierten Systemen durch ihre verteilte Struktur jedoch aufwendiger in der Bereitstellung von Diensten zu realisieren. Leider können zudem durch die Konzeption aus einer zentralen Sicht (wie in Unterabschnitt 2.3.2 erläutert) Anforderungen an den Datenschutz und die Datensicherheit in der Regel nicht durchgesetzt werden, ohne die Zielfunktionalitäten zu verlieren (siehe z. B. [AZH18, S. 51]).

Alle Systemtypen besitzen Vor- und Nachteile. Die in Unterabschnitt 2.3.5 vorgestellte Gemeinschaftshierarchie für die Aggregation von Energieverbrauchsinformationen zeigt hier einen interessanten Lösungsweg auf. Auf Basis dieser Struktur können Datenschutz- und Datensicherheitsanforderungen präzise eingestellt werden. Im Gegensatz zu den ebenfalls verteilten Edge-/Fog-Computing-Ansätzen des Energiemanagements wird hierbei eine Nutzer-zentrierte Konzeption verwendet.

### 3.3 Ein Zielsystem zur Realisierung der Vision

Aufgrund der vielen Abhängigkeiten, Verknüpfungen und der Begrenztheit bestehender Lösungen soll das Thema der Energiemanagementsysteme in verteilten Umgebungen in dieser Arbeit von einem **allgemeineren Nutzer-zentrierten Ansatz** aus betrachtet werden. Ziel ist es, ein **universelles und flexibles Systemkonzept zum Energiemanagement in verteilten Umgebungen** zu entwickeln, mit dem möglichst alle Aspekte der zuvor beschriebenen Vision bedient werden können.

Den zuvor beschriebenen Abhängigkeiten soll durch Trennung des Systemkonzepts in einzelne Schichten begegnet werden. Dadurch soll es ermöglicht werden, Techniken verschiedener Schichten frei miteinander zu kombinieren. Wie in Abbildung 3.1 dargestellt, besteht das dazu entworfene Schichtenmodell aus drei Schichten.

- Die untere Schicht ist für die **Einbindung von Geräten** zuständig. Hier sollen prinzipiell alle Techniken eingesetzt werden können, die es erlauben Informationen über Geräte bereitzustellen oder Geräte in ihrem Verhalten mit Hilfe von Steuerungsbefehlen zu beeinflussen. Das Ziel ist, *alle* Geräte unabhängig von ihren Eigenschaften integrieren zu können. Da sich die Geräte nicht nur in ihren Eigenschaften, sondern auch in ihren Kommunikationsmöglichkeiten unterscheiden, sollen im Rahmen dieser Arbeit verschiedene Techniken betrachtet werden. Einen Schwerpunkt stellen dabei

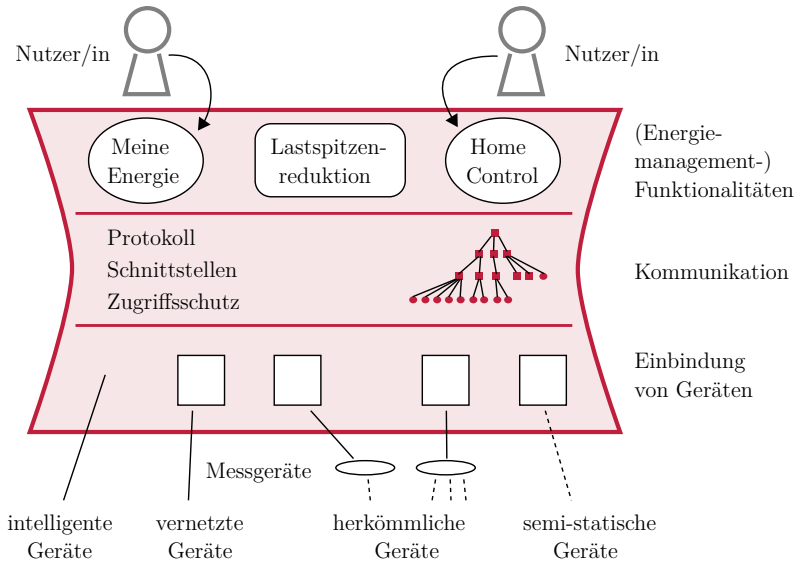


Abbildung 3.1: Schichtenmodell zur Auftrennung von Energiemanagementsystemen in verschiedene unabhängige Schichten. Durch die Konzeption nach diesem Schichtenmodell soll die Möglichkeit geschaffen werden, dass Funktionalitäten und Techniken verschiedener Schichten frei miteinander kombiniert werden können. Die untere Schicht bindet Geräte über verschiedene Techniken (Rechtecke) ein. Die Daten der Geräte werden über die mittlere Schicht der Kommunikation an die obere Schicht der Funktionalitäten vermittelt. Hier werden beispielhaft drei verschiedene Anwendungen zum Energiemanagement mit und ohne Nutzereinbindung implementiert.

Techniken dar, die eine dauerhafte Kommunikationsverbindung zu den Geräten nutzen. Weiterhin sollen auch Techniken auf Basis der Signalverarbeitung und -erkennung betrachtet werden, die es erlauben bestehende Geräte ohne Kommunikationsschnittstellen einzubinden. Damit diese untere Schicht gleichberechtigt durch alle diese Techniken vertreten werden kann, soll ein einheitlicher Funktionsumfang zur Einbindung von Geräten definiert werden.

- Die mittlere Schicht ist für die **Kommunikation** verantwortlich. Aufbauend auf der o.g. Schicht, der Einbindung von Geräten, soll eine Konnektivität und ein Austausch von Daten zwischen den Geräten und den höheren Schichten im Sinne einer Middleware (siehe Unterabschnitt 2.3.3) ermöglicht werden. Die Schicht organisiert diesen Datenaustausch transparent und über Geräte hinweg. Dabei ist der Kommunikationsbedarf hoch, da sich die Geräte verteilt in den jeweiligen Umgebungen befinden. Damit jedoch auch der Datenschutz und die Datensicherheit gewährleistet werden, sollen Daten

möglichst lokal verarbeitet werden. Hierzu bedarf es lokaler Verarbeitungsinstanzen in den jeweiligen Umgebungen.

- Die obere Schicht soll die Ausführung von eigentlichen **Funktionalitäten** ermöglichen. Die Funktionalitäten sollen dabei durch Software-Anwendungen realisiert werden, die von den lokalen Verarbeitungsinstanzen ausgeführt werden und auf den bereitgestellten Daten der tieferen Schichten aufsetzen. Einzelne Optimierungsansätze sollen dabei jeweils durch eigene Anwendungen implementiert werden. Durch dieses Prinzip sollen mehrere Optimierungsansätze parallel ermöglicht werden. Um größere Anteile des Optimierungspotentials in einer Umgebung heben zu können, sollen Anwendungen sowohl zur automatisierten Energieverbrauchsoptimierung, aber auch zum Nutzer-unterstützenden Energiemanagement realisiert werden.

Das Systemkonzept soll in Hinblick auf moderne Konzepte der Kommunikationstechnik, der Signalverarbeitung und der Anwendungsentwicklung entworfen werden. Dabei sollen die Grundideen des Internet der Dinge (siehe Abschnitt 2.3) mit dem Ziel beachtet werden, offene erweiterbare Systeme zu gestalten.

Als Basis für das Systemkonzept sowie für die spätere Überprüfung der Zielerreichung werden nachfolgend Ziele und Anforderungen definiert. In Unterabschnitt 3.3.1 werden zunächst Zielfunktionalitäten aus den Beweggründen aus Abschnitt 2.1 und 2.2 abgeleitet, die durch das Systemkonzept unterstützt werden sollen. Die Geräteeinbindung soll zudem die in Unterabschnitt 3.3.2 benannten Geräte und Techniken unterstützen. In Unterabschnitt 3.3.3 und 3.3.4 werden Anforderungen definiert. Diese betreffen insbesondere die Kommunikation und die zugrundeliegende Systemstruktur.

### 3.3.1 Zielfunktionalitäten

Im Folgenden sollen Funktionalitäten identifiziert werden, die von zukünftigen Energiemanagementsystemen grundsätzlich unterstützt werden sollen. Die Funktionalitäten wurden hierzu auf wenige Zielfunktionalitäten heruntergebrochen, um eine repräsentative Auswahl zu schaffen. Dabei wurden bewusst verschiedene Maßnahmen zu einer Funktionalität zusammengefasst. Anhand dieser wichtigsten und möglichst verschiedenartigen Funktionalitäten soll sichergestellt werden, dass das Systemkonzept grundsätzlich die aus den Beweggründen entwickelten Ziele der verschiedenen Stakeholder bedienen kann.

Die folgenden Funktionalitäten sind aus den in Unterabschnitt 2.1.4 beschriebenen Maßnahmen zum Energiemanagement abgeleitet. Die ersten zwei Funktionalitäten beziehen sich auf

Maßnahmen, die dem Ansatz der Informationsbereitstellung zur Erhöhung der Verbrauchstransparenz nach Abschnitt 2.2 zuzuordnen sind. Sie wurden aus den dort beschriebenen Anforderungen und Systemen sowohl für Privathaushalte als auch für Unternehmen und Einrichtungen abgeleitet und unterscheiden sich insbesondere in den adressierten Zielgruppen.

- Visualisierung von **Verbrauchsinformationen von Geräten im Verlauf über die Zeit** für Energiemanagerinnen und -manager, sowie für Eigentümerinnen und Eigentümer bzw. für Mieterinnen und Mieter einer lokalen Umgebung, die an einer Energiekostenreduzierung interessiert sind. Damit sollen die Maßnahmen zum „Gerätetausch“ sowie zur „Identifikation von versteckten Problemen“ aus Unterabschnitt 2.1.4 ermöglicht werden.
- Darstellung von **Informationen zu einzelnen Gerätenutzungen**, die durch eine Nutzer-Aktion ausgelöst wurden. Dies soll das Ziel adressieren, dass Nutzerinnen und Nutzer von lokalen Geräten „eigene Handlungen [überprüfen und ggfs.] anpassen“ können. Dazu sollen bedarfsweise einzelne Handlungen direkt mit den entstandenen Kosten verknüpft und der Nutzerin bzw. dem Nutzer zurückgemeldet werden.

Die folgenden Funktionalitäten sind dem zweiten Ansatz der automatisierten Steuerung von Geräten zuzuordnen. Sie sind ebenfalls aus den Maßnahmen aus Unterabschnitt 2.1.4 abgeleitet. Die Steuerung der Geräte unterscheidet sich dabei anhand der Art der Zuordnung:

- Steuerung **im Sinne einer direkten Verknüpfung**. Hier wird auf ein Ereignis oder eine Anforderung direkt mit einer Steuerung eines oder mehrerer Geräte reagiert. Dazu muss das Ereignis oder die Anforderung direkt in einen Steuerbefehl überführbar sein, z. B. durch eine binäre Ausprägung (Ein/Aus). Damit wird u. a. die Maßnahme zur gegenseitigen „Abstimmung von Geräten“ oder der Abstimmung von Geräten bei der An- bzw. Abwesenheit von Menschen adressiert (siehe Unterabschnitt 2.1.4). Das Ereignis kann dabei z. B. die Erkennung einer Nutzeranwesenheit (Ja/Nein) oder einer Statusmeldung eines Gerätes (Ein/Aus) sein. Auch die „Steuerung von Geräten aufgrund eines Direkteingriffs“ (siehe Unterabschnitt 2.1.4) soll mit dieser Funktionalität berücksichtigt werden. Die externe Anforderung (Anforderung: An/Anforderung: Aus) erfolgt dabei als Steuerung eines spezifischen Geräts.
- Steuerung **auf Basis von Schwellenwerten oder Regeln**. In Erweiterung der vorherig genannten Funktionalität sollen auch kontinuierliche Werte als Eingaben dienen können, also z. B. variable Tarife oder die prognostizierte Viertelstundenleistung. In der Regel werden hierbei Geräte und Ereignisse nicht direkt verknüpft, sondern individuell pro Gerät mittels Schwellenwerten oder mathematischen Regeln abgebildet.

Damit lässt sich die „Steuerung von Geräten aufgrund von variablen Tarifen“ oder zur „Lastspitzenreduktion“ (siehe Unterabschnitt 2.1.4) realisieren.

- Abstimmung von Geräten untereinander **anhand von Optimierungsalgorithmen**. Dabei werden neben Geräten auch Speicher- und Erzeugungsanlagen zur Deckung eines bestimmten Energiebudgets untereinander abgestimmt. Ziel ist die Maßnahme der „Gerätesteuerung zur Eigenverbrauchsoptimierung bei Prosumern“.

Zusätzlich soll eine Auswahl weiterer Funktionalitäten abgeleitet werden, die stellvertretend für die in Abschnitt 2.3 benannten Themenfelder jenseits des Energiemanagements stehen:

- Smart Home mit **Visualisierung und Steuerung von Geräten** über das Smartphone sowie einer **Szenensteuerung**, die den Betriebsstatus mehrerer Geräte gleichzeitig einstellt. Das Ziel ist es, den Bewohnerinnen und Bewohnern einen höheren Komfort zu ermöglichen (siehe auch Unterabschnitt 2.1.3).
- Prädiktive Instandhaltung anhand einer **Überwachung der Messdaten von Geräten**. Dabei soll geprüft werden, ob sich die Messwerte innerhalb zuvor definierter Toleranzen befinden. Zusätzlich kann bei Über- bzw. Unterschreitung eine Alarmmeldung ausgegeben werden.
- **Sicherheitsabschaltungen** in Abhängigkeit einer Zeitdauer oder während der Abwesenheit von Bewohnerinnen und Bewohnern z. B. zur Abschaltung von Heizquellen oder für die nachträgliche Meldung bzw. automatisierte Schließung von nicht geschlossenen Türen oder Dachfenstern. Damit wird der Beweggrund des „ruhigen Gewissens“ aus Unterabschnitt 2.1.3 adressiert.

### 3.3.2 Ziele für die Geräteeinbindung

Wie in der Vision (siehe Abschnitt 3.1) für die Verwirklichung zukünftiger Energiemanagementsysteme verdeutlicht, ist die Einbindung von möglichst vielen unterschiedlichen Geräten wünschenswert. Geräte können sich dabei in sehr vielen Dimensionen voneinander unterscheiden, wie in der Leistungsaufnahme, dem Einsatzzweck, dem Umfeld und weiteren. Für die hier gewünschte Einbindung von Geräten sollen diese anhand ihrer Kommunikationsanbindung und ihres Kommunikationsbedarfs unterschieden und entsprechende Techniken für die folgenden Klassen entwickelt werden:

- *Intelligente Geräte*, die über eine vollständig kompatible Kommunikationsanbindung verfügen. Die Kompatibilität bezieht sich darauf, dass alle benötigten Fähigkeiten unterstützt werden,

- *Vernetzte Geräte*, die über existierende Kommunikationslösungen, wie Feldbusse oder Netzwerkprotokolle eingebunden werden können, so z.B. Computer, Router, Messgeräte.
- *Semi-statische Geräte*, die permanent im Einsatz sind (Dauerläufer) oder andere Geräte, deren Verhalten voraussagbar ist, wie z. B. eine nicht geregelte und durchgängig betriebene Heizungspumpe.
- *Herkömmliche Geräte*, die bisher nicht vernetzt sind und deren Verhalten mittels zusätzlicher Sensoren erkannt werden kann.

Damit Geräte durch die Funktionalitäten möglichst auf die gleiche Weise angesprochen werden können, bedarf es der Harmonisierung eines **(einheitlichen) Funktionsumfangs** zur Interaktion mit den Geräten. Diese sollte sowohl den Abruf von Daten als auch die Beeinflussung der Geräte abdecken.

Zur Erfüllung der Funktionalitäten aus Unterabschnitt 3.3.1 sollen für jedes Gerät der Betriebsstatus, der Energieverbrauch über die Zeit sowie optional eine Steuerungsfunktion zur Beeinflussung des Betriebsstatus berücksichtigt werden. Da der Betriebsstatus sowie auch der Energieverbrauch nicht über jede Einbindungstechnik direkt erfasst werden können, bedarf es Funktionen zur Erkennung bzw. zur Nachbildung dieser Daten.

### 3.3.3 Nichtfunktionale Anforderungen

Im Folgenden werden Anforderungen aus den verschiedenen Aspekten der Vision für zukünftige Energiemanagementsysteme (siehe Abschnitt 3.1) abgeleitet, die in das später dargestellte Systemkonzept einfließen sollen. Da viele der Anforderungen das Systemkonzept sowie auch die Implementierung im Ganzen betreffen, werden sie dort nur im Einzelfall benannt. In der späteren Evaluation soll anhand dieser Anforderungen untersucht werden, ob die Implementierung und das zugrundeliegende Systemkonzept die Aspekte der Vision abdecken.

Zunächst werden nichtfunktionale Anforderungen dargestellt, die die zu realisierenden Eigenschaften des Zielsystems beschreiben.

#### Anforderung NF.1: Skalierungsfähigkeit

Das Ziel, Energie zu managen, stellt eine kooperative Aufgabe zwischen verschiedenen Systemen getrennt durch unterschiedliche Distanzen dar. Energiemanagementlösungen könnten in Zukunft großflächige Verbreitung finden. Das umfassendste Vernetzungsziel wäre ein Zusammenschluss, der sich – wie die zugrundeliegenden Energieübertragungsnetze – über

gesamte Kontinente erstreckt. Im Fall des europäischen Verbundnetzes -- verwaltet durch den Verband Europäischer Übertragungsnetzbetreiber für Strom (engl. European Network of Transmission System Operators for Electricity, ENTSO-E) – betrifft dieses 36 Länder (siehe [ENT18]).

Daraus ergibt sich die Anforderung, dass eine hohe Skalierungsfähigkeit für das System insbesondere im Vergleich zu zentralisierten Systemen (siehe Unterabschnitt 2.3.2) von entscheidender Bedeutung ist.

#### **Anforderung NF.2: Datenschutz und Datensicherheit**

Bei jeglicher Kommunikation und Datenverarbeitung müssen Datenschutz- und Datensicherheitsanforderungen beachtet werden, damit sensible Daten, wie z. B. Betriebsgeheimnisse und die Privatsphäre gewahrt werden können. Auch Schnittstellen, die Steuerungsfunktionen zur Verfügung stellen, müssen vor Fremdzugriff geschützt werden.

Ziel dieser Anforderung ist es, dass die Nutzerinnen und Nutzer jederzeit die Kontrolle über das System und ihre eigenen Daten behalten. Deshalb sollen (wie bereits in der Einleitung zu Abschnitt 3.3 erwähnt) die Daten möglichst auf Geräten in den lokalen Umgebungen verarbeitet werden, in denen sie entstanden sind oder in denen sie benötigt werden. Dies soll bewirken, dass Daten die lokalen Umgebungen nur dann verlassen und damit an Dritte übergeben werden, wenn dies erforderlich und sinnvoll ist.

#### **Anforderung NF.3: Ausfallsicherheit**

Das Gesamtsystem soll eine möglichst hohe Ausfallsicherheit bieten. Dies kann zukünftig von hoher Relevanz sein, wenn mehr und mehr kritische Infrastrukturen mit dem System verbunden werden. Beispielsweise sollte bei einem Beleuchtungssteuerungssystem die Steuerung der Beleuchtung noch möglich sein, auch wenn keine Verbindung zum Internet besteht. Dies gilt im Kontext des Energiemanagements, des Smart Home oder der Vernetzung im Sinne der Industrie 4.0 für so gut wie jede technische Einrichtung.

#### **Anforderung NF.4: Fehlertoleranz und Robustheit**

Ähnlich wie bei der Ausfallsicherheit sollen alle Komponenten weitestgehend tolerant auf Fehler reagieren. Beispielsweise soll sichergestellt werden, dass Netzwerkverbindungen automatisch wieder hergestellt sowie Ressourcen in regelmäßigen Abständen auf ihre Verfügbarkeit überprüft werden.

**Anforderung NF.5: Erweiterungsfähigkeit**

Damit mehr Energiemanagement-Funktionalitäten und Geräteeinbindungen für das System entstehen können, sollte für Drittanbieter die Möglichkeit bestehen, das System erweitern zu können.

Die Erweiterungsmöglichkeiten sollten möglichst alle drei Schichten umfassen, mindestens sollen jedoch sowohl Techniken zur Geräteeinbindung als auch Anwendungen zum Energiemanagement von Drittanbietern hinzugefügt werden können.

**Anforderung NF.6: Implementierungsaufwand**

Der Implementierungsaufwand von Techniken zur Geräteeinbindung als auch von Anwendungen zum Energiemanagement sollte so weit wie möglich gering gehalten werden. Da eine generelle Bewertung des Implementierungsaufwands nur schwer möglich ist, sollen bei der Analyse die Maßnahmen zur Verringerung des Implementierungsaufwands betrachtet werden.

**Anforderung NF.7: Installierbarkeit und Aktualisierung**

Weiterhin soll die Installation und Aktualisierung des Basissystems sowie aller weiteren Komponenten mit geringem Aufwand erfolgen können.

Das System soll dabei auf möglichst vielen Gerätearten und Betriebssystemen installierbar sein.

### 3.3.4 Funktionale Anforderungen

In Erweiterung zum vorherigen Unterabschnitt 3.3.3 werden nachfolgend die funktionalen Anforderungen abgeleitet. Die Anforderungen begründen sich größtenteils durch das Schichtenmodell und die damit verbundenen Funktionen (vgl. Einleitung zu Abschnitt 3.3). Einige Anforderungen werden zusätzlich aus den Zielfunktionalitäten (aus Unterabschnitt 3.3.1) motiviert.

**Anforderung F.1: Kapselung der Funktionen nach dem Schichtenmodell**

Die Trennung in einzelne Schichten, wie in Abschnitt 3.3 beschrieben, soll durch eine Kapselung der Funktionen realisiert werden. Unter dem Begriff der Kapselung ist zu verstehen, dass Funktionen vom Basissystem abgespalten werden und beispielsweise als eigenständige Software-Anwendungen, Module, Plugins oder andere Arten der Erweiterung implementiert werden. Diese Kapselung soll sowohl bei Techniken zur Einbindung der Geräte, als auch bei Funktionalitäten zum Energiemanagement angewendet werden.



Mit der Kapselung wird das Entwurfsprinzip der Separation of Concerns (engl. sinngemäß für Trennung der Anliegen) umgesetzt. Dies bedingt, dass verschiedene Aufgaben in eigenen Teillösungen umgesetzt werden, um die verschiedenen Anliegen getrennt voneinander betrachten zu können. (vgl. [Ell14, Kapitel 5]) Die Kapselung verbessert zudem die Erweiterungsfähigkeit durch Drittanbieter, wie in Unterabschnitt 3.3.3 als Ziel benannt. Des Weiteren fördert sie eine Wiederverwendbarkeit der Funktionen.

Die Software-Anwendungen sollen (z. B. vom Eigentümer) separat installiert, aktualisiert und konfiguriert werden können (siehe Unterabschnitt 3.3.3, Bereitstellung und Aktualisierung).

### **Anforderung F.2: Transparente Kommunikation**

Damit die gekapselten Funktionen miteinander kommunizieren und interagieren können, bedarf es der Schnittstellen zwischen den Schichten. Aufgrund der Verteilung der einzubindenden Geräte in den Umgebungen, bedarf es zudem einer Datenkommunikation über Geräte hinweg.

Um die Trennung in Schichten zu realisieren, soll die Kommunikation transparent erfolgen. Damit soll erreicht werden, dass der Kommunikationsaustausch von Elementen anderer Schichten weder bemerkt noch berücksichtigt werden muss.

Die Zielfunktionalitäten (siehe Unterabschnitt 3.3.1) bedingen, dass verschiedene Komponenten miteinander kommunizieren müssen. Insbesondere sollen folgende Elemente aufeinander zugreifen können:

- Die Energiemanagement-Anwendungen sollen auf die Daten der eingebundenen Geräte zugreifen können.
- Die Energiemanagement-Anwendungen sollen auch Steuerbefehle aussenden können, um Geräte (wenn möglich) in ihrem Verhalten zu beeinflussen.
- Energiemanagement-Anwendungen sollen auf Funktionen und Daten anderer Energiemanagement-Anwendungen zugreifen können, um mit diesen zu kooperieren.
- Auch die Einbindung von globalen Informationen soll ermöglicht werden. Durch diese sogenannte Cloudanbindung soll auch von Datenbeständen zentraler Cloud-Systeme profitiert werden.

### **Anforderung F.3: Latenzanforderungen für die Kommunikation**

Verschiedene Anwendungsfälle besitzen unterschiedlich hohe Anforderungen an die zeitliche Verzögerung durch die Kommunikation. Diese Verzögerung wird auch Latenz genannt. Die

Latenzanforderungen reichen bei zeitkritischen Funktionen bis hin zu Echtzeitanforderungen<sup>3</sup> z. B. im Bereich der Industrie 4.0 (siehe Unterabschnitt 2.3.1).

Mit der Anforderung an eine möglichst geringe Latenz, wird sichergestellt, dass möglichst viele Anwendungsfälle realisiert werden können.

#### **Anforderung F.4: Begrenzung der Kommunikation / Zugriffsschutz**

In Unterabschnitt 3.3.3 (Anforderung an Datenschutz und Datensicherheit) wurde bereits beschrieben, dass es möglich sein soll, Datenschutz- und Datensicherheitsanforderungen zu beachten. Konkrete Datenschutz- und Datensicherheitsanforderungen müssen immer dann durchgesetzt werden, wenn Daten prinzipiell die lokale Umgebung verlassen können.

Hierzu soll die Möglichkeit bestehen, die Datenweitergabe auf jedem Kommunikationsweg einschränken zu können. Genauso soll ein Zugriffsschutz für bereitgestellte Funktionalitäten ermöglicht werden.

#### **Anforderung F.5: Konfiguration durch automatische Verknüpfung / Auto Discovery**

Daten (von Geräten) und Anwendungen müssen miteinander verknüpft werden, damit die Software-Anwendungen ihre Aufgaben erfüllen können. Eine solche Verknüpfung kann durch eine Nutzerinteraktion konfiguriert werden. Im Idealfall soll die Verknüpfung der Elemente der Anwendungsschicht und der Geräteeinbindung, die miteinander kooperieren, eigenständig geschehen. Dazu sollen die Anwendungen auch in der Lage sein, Daten und Funktionen zu identifizieren, die für sie von Interesse sind. Eine Nutzerintervention soll nur in Ausnahmen erforderlich sein.

Mit dieser Anforderung soll auch die Mehrfachnutzung von Ressourcen avisiert werden, die eine der Grundideen des Internet der Dinge ist.

#### **Anforderung F.6: Nutzeroberflächen**

Damit Nutzerinnen und Nutzer mit den Energiemanagement-Anwendungen interagieren können, sollen Nutzeroberflächen u.A. auf Smartphones, Tablets und PCs zur Verfügung gestellt werden können.

Die Oberflächen sollen mittels eines Entwurfsschemas, wie z. B. Modell-View-Control (engl. für die drei Komponenten Datenmodell, Anzeige und Controller, MVC), entworfen werden können.

---

<sup>3</sup>Unter dem Begriff Echtzeit wird die Garantie verstanden, dass eine Aufgabe in einer gegebenen Zeit erledigt werden kann, die aus Sicht der Anwendung unmerklich ist (vgl. [Sch05, S. 40]).

## 4 Systemkonzept zum Energiemanagement in verteilten Systemen

In diesem Kapitel wird das **Systemkonzept für Energiemanagementsysteme** vorgestellt. In Abschnitt 3.3 wurde bereits erläutert, dass als Lösungsansatz zur Überwindung der Defizite vorhandener Lösungen ein Schichtenmodell bestehend aus drei Schichten zugrunde gelegt wird. Mit diesem Schichtenmodell soll ermöglicht werden, unterschiedliche Techniken und Funktionalitäten frei miteinander zu kombinieren. Im Folgenden wird die Konzeption der jeweiligen Schichten vorgestellt.

Die untere Schicht ermöglicht die **Einbindung von Geräten**. Wie in der Einleitung (siehe Kapitel 1) dargestellt, werden in dieser Arbeit technischen Systeme betrachtet, die elektrische Energie direkt aus dem Energieversorgungsnetz beziehen oder lokal erzeugen und unter dem Begriff *Geräte* zusammengefasst. Darunter sind neben Haushaltsgeräten, Unterhaltungselektronik, Bürogeräten sowie IT-Geräten auch Industriemaschinen sowie Anlagen der technischen Gebäudeausstattung (TGA) zu verstehen.

Wie in Kapitel 3 gefordert, sollen *alle* Geräte trotz ihrer Verschiedenartigkeit in das System eingebunden werden können. Die Geräte unterscheiden sich dabei u. a. in der Fähigkeit der kommunikationstechnischen Einbindung in die Energiemanagementsysteme. Hierzu werden die Geräte in dieser Arbeit in Klassen eingeordnet. Zunächst wird zwischen *vernetzten* Geräten, die dem System ihre Betriebszustände direkt bereitstellen können, und *nicht-vernetzten* Geräten unterschieden. Kann ein Gerät mit Kommunikationsanbindung seinen Betriebszustand nicht über diese bereitstellen, wird dieses nachfolgend ebenfalls als nicht-vernetztes Gerät behandelt.

Die Einbindung von vernetzten Geräten wird in Abschnitt 4.1 beschrieben. Dabei wird herausgearbeitet, welche Daten und Funktionen zum Eingriff in den Energieverbrauch von einem vernetzten Gerät zur Verfügung gestellt werden sollten. Hier ist zu berücksichtigen, dass bereits vorhandene vernetzte Geräte ggfs. nicht alle notwendigen oder wünschenswerten Daten und Funktionen bereitstellen. Aus diesem Grund wird eine Gerätemodellierung zur Nachbildung dieser Daten und Funktionen mit Hilfe der Signalverarbeitung vorgeschlagen.

Aus diesem Ansatz wird ein Verfahren abgeleitet, mit dem aus dem Betriebsstatus der Energieverbrauch eines Geräts abgeschätzt werden kann.

In Abschnitt 4.2 wird die Einbindung von nicht-vernetzten Geräten erläutert. Da zu nicht-vernetzten Geräten keine direkte Kommunikationsverbindung besteht, ist folgerichtig kein direkter Datenaustausch möglich. Dennoch sollen auch diese Geräte im Energiemanagementsystem berücksichtigt werden, da diese in bestehenden Haushalten und vermutlich auch in vielen Unternehmen erfahrungsgemäß noch vorherrschen. Im Sinne der *digitalen Repräsentation*, wie sie im Internet der Dinge (siehe Abschnitt 2.3) zur Abbildung nicht dauerhaft vernetzter Gegenstände verwendet wird, sollen reale Geräte in Datenobjekten nachgebildet werden. Bei dieser IoT-Gerätetrepräsentation muss zwischen der Qualität der Repräsentation und den anfallenden Kosten, wie z. B. für die Installation zusätzlicher Sensoren, abgewogen werden. Ohne zusätzliche Messdaten ist eine solche Repräsentation nur bei Geräten mit statischem Verhalten oder Geräten mit planbarem Verhalten mit geringem Fehler möglich. Mit zusätzlichen Messdaten kann die IoT-Gerätetrepräsentation je nach Anwendungsfall weiter verbessert werden. Zusätzlich kann durch geeignete Sensoren auf den Betriebsstatus der Geräte geschlossen werden. Für diese Erkennungsaufgabe werden regelbasierte sowie statistische Verfahren vorgestellt. Unter bestimmten Voraussetzungen kann mit Hilfe eines Sensors der Betriebsstatus gleichzeitig mehrerer Geräte abgebildet werden. Dadurch könnten die anfallenden Kosten für eine Anbindung bestehender nicht-vernetzter Geräte begrenzt werden.

Die mittlere Schicht erfüllt die Aufgabe der **Kommunikation** zwischen der unteren und der oberen Schicht. Diese wird näher in Abschnitt 4.3 erläutert. Dabei ist die Herausforderung zu beachten, dass Kommunikationspartner in unterschiedlichen, nicht direkt verbundenen, Umgebungen angesiedelt sein können. Damit eine Kommunikation ermöglicht und dem Datenschutz bzw. der Datensicherheit – wie in Abschnitt 3.3 gefordert – entsprochen werden kann, wird eine hierarchische Struktur für die Kommunikation zugrunde gelegt. Dadurch wird ermöglicht, dass zwei Kommunikationspartner durch vermittelnde Geräte miteinander kommunizieren können, ohne, dass a) auf die Position oder Adresse des gegenüberliegenden Kommunikationspartners und b) durch einzelne Datensätze auf sensible Daten geschlossen werden kann. Als Beispiel könnten mehrere Endverbraucher ihre elektrische Energieerzeugung sowie ihren Energieverbrauch aufeinander abstimmen und damit jeweils wirtschaftliche Vorteile erzielen. Durch die hierarchische Kommunikationsstruktur soll dabei unter anderem ermöglicht werden, dass die Endverbraucher nicht auf die Verhaltensweisen der jeweils anderen Partei schließen können.

Die Zielsetzung an das Energiemanagement verlangt auch die Möglichkeit, dass verschiedene Geräte gemeinsame Aufgaben übernehmen. Da die Daten und Funktionen der Geräte

(untere Schicht) nicht in einem einzigen zentralen System zusammengeführt werden sollen, müssen die Entscheidungen über die anstehenden Aufgaben dort entschieden werden, wo die Daten vorliegen also z. B. an den jeweiligen Datenquellen. Auf Grundlage der hierarchischen Kommunikationsstruktur wird deshalb ebenfalls in Abschnitt 4.4 ein Konzept für dezentrale Anwendungen vorgestellt. Dies soll erlauben, dass eine einzige Anwendung über mehrere Geräte hinweg ausgeführt werden und dabei weiterhin eindeutige Entscheidungen treffen kann.

Die **Funktionalitäten** des Energiemanagements werden in Abschnitt 4.4 für Nutzer-unterstützte Ansätze und in Abschnitt 4.5 für automatisierte Ansätze näher betrachtet. Nach dem Schichtenmodell sind solche Funktionalitäten der oberen Schicht zugeordnet. Im Rahmen des Systemkonzepts sollen die Funktionalitäten als verteilte Anwendungen nach dem beschriebenen Anwendungskonzept implementiert werden. Dazu wird anhand von Diagrammen die Struktur der jeweiligen Funktionalitäten erläutert.

In Abschnitt 4.6 werden zusätzlich Funktionalitäten jenseits des Energiemanagements betrachtet. Diese Funktionalitäten zeigen die Ähnlichkeit der Themengebiete und sollen verdeutlichen, dass das Systemkonzept nicht auf Energiemanagement-Anwendungen beschränkt ist.

Erste Ansätze dieses Systemkonzepts werden in den Veröffentlichungen [2], [3] und [4] vorgestellt. In [5] wird zusätzlich das Konzept der verteilten Anwendungen beschrieben.

## 4.1 Einbindung von vernetzten Geräten

Dieser Abschnitt beschäftigt sich mit der Einbindung von vernetzten Geräten. Das Ziel ist es, eine einheitliche Technik-unabhängige Schnittstelle zwischen dem Energiemanagementsystem und den Geräten zu entwerfen.

In Unterabschnitt 4.1.1 wird der notwendige Funktionsumfang dieser Schnittstelle beschrieben. Um Geräte einzubinden, die bestehende Kommunikationstechniken nutzen, ist eine Adaptierung mit Hilfe einer zusätzlichen Kompatibilitätsschicht erforderlich. Diese Kommunikationstechniken können jedoch Einschränkungen bezogen auf den geforderten Funktionsumfang unterliegen. In Unterabschnitt 4.1.2 wird deshalb erläutert, wie mit diesen Einschränkungen umgegangen werden soll. Dazu wird in Unterabschnitt 4.1.3 eine Geräte-modellierung zur Nachbildung von Geräten vorgestellt. Aufbauend auf diesem Ansatz wird in Unterabschnitt 4.1.4 eine Technik zur Nachbildung fehlender Informationen beschrieben. In Unterabschnitt 4.1.5 werden Geräte in verschiedene Arten aufgeteilt, abhängig davon, auf

welche Weise sie in ihrem Verhalten beeinflusst werden können. Abschließend wird in Unterabschnitt 4.1.6 die Nachbildung einer Funktionalität zur Beeinflussung des Betriebsstatus beschrieben.

### 4.1.1 Notwendiger Funktionsumfang von vernetzten Geräten

Damit ein vernetztes Gerät direkt in das Energiemanagementsystem eingebunden werden kann, sollen die nachfolgenden *Daten* und *Funktionen* bereitgestellt werden können. Die hervorgehobenen Begriffe kennzeichnen jeweils die beschriebenen Eigenschaften, auf die nachfolgend Bezug genommen werden wird.

- Als erster Schritt muss ein Gerät dem Energiemanagementsystem seine **Existenz** bekannt geben, damit das System anschließend auf dessen restliche Daten und Funktionen zugreifen kann. Für eine verifizierte Kopplung ist zusätzlich ggfs. eine Nutzeraktion notwendig.
- Bei dieser Bekanntgabe kann das Gerät auch zusätzlich **Metadaten** zur eigenen Beschreibung mitliefern. Relevante Daten könnten eine eindeutige Identifikationskennung zur Wiedererkennung, eine Produktbezeichnung mit Herstellerangabe sowie ggfs. ein benutzerdefinierter Name und eine Ortangabe sein.
- Das Gerät muss das System über seinen aktuellen **Betriebsstatus** unterrichten (siehe Forderung Unterabschnitt 3.3.2). Beispielhafte Ausprägungen könnten sein: „An“, „Aus“, „Standby“, „Vor-Heizen“, „In Produktion“ (bei Industriemaschinen). Sofern das Gerät aus mehreren Komponenten besteht, ist auch der Betriebsstatus der jeweiligen Einzelkomponenten von Interesse. Eine Heizungsanlage sollte also den Betriebsstatus seiner Komponenten, wie beispielsweise Brenner und Umwälzpumpen bereitstellen. Ein textueller Betriebsstatus gibt jedoch oft nur grob den Operationsmodus des Gerätes wieder. Aus diesem Grund kann es sinnvoll sein, wenn das Gerät auch interne Statusdaten zu Verfügung stellt, damit der Zustand des Geräts vollständig charakterisiert werden kann. Führungs-, Regel- und Steuergrößen einer Geräte-internen Regelung sind Beispiele für solche internen Statusdaten.
- Weiterhin ist der durch den Einsatz des Gerätes verursachte Energieverbrauch von besonderem Interesse (siehe Forderung Unterabschnitt 3.3.2). Das Gerät kann dem Energiemanagementsystem diese und weitere **Sensordaten** als fortlaufende Datenreihen bereitstellen. Dazu gehören neben dem Energieverbrauch auch Größen, die den Betriebsstatus oder den Energieverbrauch beeinflussen können.

- Damit im Bedarfsfall eine automatisierte Energieverbrauchsoptimierung ermöglicht werden kann, muss möglichst eine optionale Funktionalität zur **Beeinflussung des Betriebsstatus** – und damit des Energieverbrauchs – angeboten werden. So könnte ein Gerät nach Aufforderung zeitweise in den Standby-Zustand versetzt werden, wenn dessen Funktion zeitweise verzichtbar oder aufschiebbar ist. Zur Berechnung einer automatisierten Energieverbrauchsoptimierung werden Prognosen über den *zukünftigen* Energieverbrauch benötigt (siehe Forderung Unterabschnitt 3.3.2). Hierzu muss ein Gerät eine Prognose seines Energieverbrauchs unter den laufenden Beeinflussungen sowie die auftretenden Folgekosten eines Eingriffs mitteilen. Folgekosten können entstehen, wenn durch eine längere Unterbrechung erneut Energie aufgewendet werden muss, damit der Prozess fortgeführt werden kann. Wird beispielsweise ein Produktionsprozess in der Schokoladenerzeugung unterbrochen, müssen zunächst die Solltemperaturen wieder erreicht werden, bevor die Fertigung fortgesetzt werden kann. In seltenen Fällen können zudem große Sachschäden auftreten, die ebenfalls durch hohe Folgekosten modelliert werden sollten. Erstarren zum Beispiel im o.g. Prozess die Schokoladenmassen, ist eine kostenintensive Reinigung bzw. der Austausch der Fertigungseinheiten erforderlich. Wird dies durch die automatisierte Energieverbrauchsoptimierung berücksichtigt, können Zustände mit (hohen) Folgekosten vermieden werden.
- Zusätzlich könnten je nach Energiemanagement-Anwendung weitere Daten oder Funktionen von Interesse sein. Da dies jedoch stark vom Anwendungsfall abhängig ist, sollen vernetzte Geräte eine **Erweiterungsmöglichkeit** aufweisen. Damit soll die Möglichkeit gegeben werden, einen Programmcode direkt auf den Geräten auszuführen und so die gewünschten zusätzliche Daten oder Funktionen auszuführen.

Geräte, die die genannten Eigenschaften aufweisen, werden im Rahmen dieser Arbeit als *IoT-Geräte* bezeichnet. Die geforderten Eigenschaften müssen bei der Realisierung in Form eines Protokolls umgesetzt werden.

### 4.1.2 Einbindung von Geräten über vorhandene Kommunikationstechniken

Es existieren bereits zahlreiche Kommunikationstechniken zum Datenaustausch mit Geräten (siehe Unterabschnitt 2.3.1), jedoch bilden die Geräte in der Regel nicht die in Unterabschnitt 4.1.1 geforderten Eigenschaften ab. Über eine *Kompatibilitätsschicht*, die diese Schnittstellen implementiert und auf die Datenstrukturen des Energiemanagementsystems abbildet, sollen Geräte mit den bestehenden Kommunikationstechniken dennoch eingebunden werden. Die Kompatibilitätsschicht kann dabei als Software-Komponente oder als eigenständiges Gerät

z. B. in Form eines sogenannten Bus-Kopplers realisiert werden. Es können auch Adapter speziell zur Einbindung eines einzelnen Geräts realisiert werden.

Bei der Einbindung von Geräten über vorhandene Kommunikationstechniken stehen jedoch in der Regel aufgrund der Beschränkung der Protokolle, der Geräte oder der Messtechnik in den Geräten nicht alle in Unterabschnitt 4.1.1 beschriebenen Daten und Funktionen zur Verfügung. Nachfolgend wird erläutert, wie mit fehlenden Daten oder Funktionen umgegangen werden soll.

- Die **Existenz** eines Gerätes, wie ggfs. auch die **Metadaten**, müssen manuell von den Anwenderinnen und Anwendern in das System eingetragen werden, falls die Kommunikationstechnik dieses nicht übernimmt. Hierzu können der Anwenderin bzw. dem Anwender Unterstützungsfunktionen bereitgestellt werden. So könnte ein Gerät durch das Scannen eines Barcodes hinzugefügt werden.
- Aufgrund der Definition der hier behandelten vernetzten Geräte (siehe Einleitung Kapitel 4), wird der **Betriebsstatus** als verfügbar und nutzbar angesehen. Geräte für die diese Forderung der Definition nicht zutrifft und die über Kompatibilitätsschichten angebunden werden, müssen deshalb mit weitergehenden Techniken ertüchtigt werden. Auf diese Techniken wird nachfolgend bei der Einbindung von nicht-vernetzten Geräten genauer eingegangen (siehe Abschnitt 4.2).
- Vielfach stehen für den Betrieb der Energiemanagement-Anwendung relevante oder interessante **Messdaten** nicht zur Verfügung. Insbesondere auf die Messung des Energieverbrauchs wird innerhalb der Geräte oft verzichtet, da dies mit zusätzlichem Hardwareaufwand verbunden ist. Im nächsten Unterabschnitt 4.1.2 wird ein Ansatz vorgestellt, mit dem Messdaten mittels einer Gerätemodellierung rekonstruiert werden können.
- Aufbauend auf diesem Ansatz wird in Unterabschnitt 4.1.6 eine Möglichkeit erläutert, wie die beschriebene Funktionalität zur **Beeinflussung des Betriebsstatus** nachgebildet werden kann.
- Die geforderte **Erweiterungsmöglichkeit** muss ebenfalls durch die Kompatibilitätsschicht implementiert werden. Dies führt jedoch durch die eingeschränkten Schnittstellen der verwendeten Kommunikationstechnik zu Beschränkungen in der möglichen Erweiterungsfähigkeit.



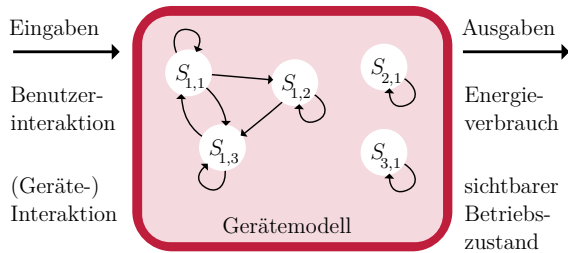


Abbildung 4.1: Gerätemodell mit internen Zuständen  $\vec{s}(t)$ , den Übergängen, die das Gerät charakterisieren, sowie Eingaben  $\vec{e}(t)$  und Ausgaben  $\vec{a}(t)$

### 4.1.3 Gerätemodellierung zur Nachbildung von Geräten

Für die Nachbildung fehlender Informationen wird nachfolgend ein allgemeines Gerätemodell konzipiert, mit dessen Hilfe die internen Abläufe eines einzelnen Gerätes nachempfunden werden können. Dieses ist in Abbildung 4.1 veranschaulicht. Das Gerätemodell beschreibt die Abhängigkeiten zwischen Eingaben  $\vec{e}(t)$ , internen Zuständen  $\vec{s}(t)$  und Ausgaben  $\vec{a}(t)$ . Die Größen variieren hier über die Zeit  $t$ . In allen nachfolgenden Formeln ist die Zeitachse dabei als diskrete Zeitachse  $t' = \frac{t}{\Delta T}$  mit dem Abtastintervall  $\Delta T$  ausgeführt.

Unter *Eingaben*  $\vec{e}(t)$ , also Größen, die die internen Zustände beeinflussen, sind demnach Nutzer-Eingaben, Messgrößen (wie z. B. eine Temperatur) oder Ereignisse (z. B. ein anderes verknüpftes Gerät wurde eingeschaltet) zu verstehen.

Die *internen Zustände*  $\vec{s}(t)$  charakterisieren vollständig das Verhalten eines Geräts (soweit modelliert). Sie entsprechen damit dem zuvor beschriebenen aktuellen Betriebsstatus. Mit Hilfe dieser Zustände und eines Zustandsmodells können jegliche Abhängigkeiten modelliert werden. Da ein Gerät aus mehreren Komponenten  $K$  bestehen und damit unabhängige Abläufe enthalten kann, werden die internen Zustände  $\vec{s}(t)$  durch mehrere unabhängige interne Zustandsgrößen  $s_k(t)$  mit  $k \in \{1, \dots, K\}$  abgebildet. Sie entstammen dem Zustandsraum  $\mathbb{S}_k$ , mit  $s_k(t) \in \mathbb{S}_k$ . Der Zustandsraum  $\mathbb{S}_k$  kann entweder aus diskreten Zuständen  $\{S_{k,1}, \dots, S_{k,P}\}$  mit  $N_k$  Ausprägungen bestehen, wie in der Abbildung 4.1 dargestellt, oder auch einen kontinuierlichen Wertebereich, z. B.  $\mathbb{R}$ , abbilden.

Wie in der Definition der vernetzten Geräte beschrieben, soll davon ausgegangen werden, dass der Betriebsstatus von vernetzten Geräten vorliegt (siehe Einleitung Kapitel 4) und der aktuelle Zustand damit vollständig (oder zumindest im Sinne der geplanten Anwendungsfälle ausreichend) durch die zur Verfügung stehenden Status- und Messdaten wiedergegeben wird.

Für die weiteren Nachbildungen in diesem Kapitel kann also angenommen werden, dass die internen Zustände durch

$$\vec{s}(t) = [z_1(t), \dots, z_R(t), m_1(t), \dots, m_V(t)]$$

abgebildet werden können, wobei  $z_r(t)$  die Statusdaten der Anzahl  $R$  inkl. des Betriebsstatus und  $m_v(t)$  die Messdaten der Anzahl  $V$  symbolisieren, die durch das Gerät bereitgestellt wurden.

Als *Ausgaben*  $\vec{a}(t)$  werden jegliche beobachtbare Größen verstanden, die durch die internen Zustandsgrößen (direkt) beeinflusst werden. Sie werden damit in diesem Modell als statistisch abhängig von den internen Zustandsgrößen modelliert. Beispiele für solche Ausgaben sind die Anzahl der produzierten Teile eines Fertigungsprozesses, ein für Menschen formulierter Betriebsstatus oder der Energieverbrauch eines Geräts.

#### 4.1.4 Nachbildung fehlender Sensordaten

Auf Basis der Gerätemodellierung (siehe Unterabschnitt 4.1.3) soll das Energiemanagementsystem fehlende Sensordaten nachbilden können. Bei den erforderlichen Sensordaten, wie dem Energieverbrauch, handelt es sich in der Regel um Ausgaben. Da die Ausgaben nach ihrer Definition in Unterabschnitt 4.1.3 abhängig von den internen Zuständen sind, können fehlende Ausgaben  $\vec{a}(t)$  aus den internen Zuständen abgeleitet werden. Hierzu genügt die Abbildung  $M_s$  mit

$$\vec{a}(t) = M_s(\vec{s}(t - \tau), \dots, \vec{s}(t)),$$

der internen Zustände  $\vec{s}$  über die Vergangenheit  $t - \tau$  bis  $t$  mit der Gedächtnislänge  $\tau \geq 0$ . Weitere Abhängigkeiten werden hier vernachlässigt. Je nach Komplexität des Gerätes und der zur Verfügung stehenden Informationen kann es sinnvoll sein für die Abbildung unterschiedliche Schätzalgorithmen einzusetzen. Die Schätzalgorithmen benötigen eine Parametrisierung, die den Zusammenhang zwischen den betrachteten Größen des Gerätes beschreibt. Diese notwendigen Parameter werden zusammenfassend als Modell bezeichnet.

Die einfachste Möglichkeit für einen Schätzalgorithmus besteht darin, jedem konkreten Betriebsstatus einen Ausgabewert zuzuordnen. Ein passendes Modell kann manuell durch eine Expertin oder einen Experten erstellt werden. Diese Methode ist aufwendig und bildet insbesondere komplexe Geräte nicht ordnungsgemäß nach. Durch eine mathematische Regression oder ein Verfahren des automatischen Lernens können diese Modelle auch automatisiert erzeugt werden. Für dieses Training ist jedoch zusätzlich eine Einzelmessung der Ausgabewerte notwendig. Solche Modelle könnten zukünftig auch über das Internet

ausgetauscht werden. Dabei ist zu beachten, dass ein solches Modell vermutlich nur auf das gleiche Gerätemodell des jeweiligen Herstellers angewendet werden kann, da unterschiedliche Geräte in ihrem Verhalten abweichen können.

Mit dieser Schätzung wird das Energiemanagementsystem unter anderem in die Lage versetzt, selbstständig fehlende Werte des Energieverbrauchs aus internen Zuständen zu ermitteln. Dieses Verfahren wird nachfolgend als *Energieverbrauchsermittlung* bezeichnet. Eine solche Abschätzung ist mit Fehlern verbunden, da der Energieverbrauch durch viele Einflussfaktoren, aber auch durch Rauschen und andere Zufallsprozesse beeinflusst wird. Auch Alterungseffekte oder einzelne defekte Baugruppen können sich auf den Energieverbrauch auswirken. Damit ist diese Schätzung nicht für Abrechnungszwecke oder ähnliche Anwendungsfälle geeignet, die eine exakte Datengrundlage benötigen. Dennoch kann eine solche Nachbildung sinnvoll sein, um beispielsweise einer Anwenderin bzw. einem Anwender die Aufschlüsselung eines Energieverbrauchs zu ermöglichen. Weiterhin könnte diese Methodik zur Ermittlung der Prognosen einzelner Geräte für eine automatisierte Energieverbrauchsoptimierung angewandt werden.

Auch Eingaben können aus den internen Zuständen nachgebildet werden, jedoch ist dies nicht immer eindeutig möglich, da hier von der Folge auf die Ursache zurück geschlossen werden muss. So können verschiedene Eingaben (zunächst) zu den gleichen internen Zuständen führen. Eine solche Nachbildung ist durch eine Modellierung der Eingaben, beispielsweise auf der Grundlage von Hidden-Markoff-Modellen (siehe [Rab89]), möglich. Dies wird im Rahmen dieser Arbeit jedoch nicht näher betrachtet.

#### 4.1.5 Beeinflussbarkeit des Betriebsstatus von vernetzten Geräten

Die Beeinflussung des Betriebsstatus ist nicht bei jedem Gerät in der gleichen Qualität möglich. Folgend sollen drei Klassen anhand ihrer Nutzung unterschieden werden:

- **Automatische Geräte** arbeiten selbstständig. Diese Geräte sollten Funktionen zu deren Beeinflussung anbieten. Eine Nutzerinteraktion findet nur zur Konfiguration der Geräte statt. Beispiele für diese Klasse sind Kühlschränke, Lüftungsanlagen und Fertigungsstraßen. Auch viele Geräte der technischen Gebäudeausstattung (TGA) sind hier aufzuzählen.
- **Halbautomatische Geräte** führen nach Aktivierung durch die Nutzerin bzw. den Nutzer ein Programm automatisch aus. Ein Beispiel für ein solches Gerät ist ein Geschirrspüler. Wurde beispielsweise ein solches Gerät durch die Nutzerin bzw. den Nutzer in Betrieb gesetzt, wird anschließend das Reinigungsprogramm bestehend

aus mehreren Heiz-, Spül- und Trocknungsphasen ausgeführt. Solche Programme sollten durch die automatisierte Energieverbrauchsoptimierung beeinflussbar sein. Beim Startvorgang könnte die Nutzerin bzw. der Nutzer zusätzlich einen Zeitpunkt angeben an dem der Programmablauf des Gerätes beendet sein sollte. Innerhalb dieses Zeitintervalls kann das System eine Optimierung durchführen.

- **Manuelle Geräte** werden direkt von der Nutzerin bzw. vom Nutzer bedient. Eine Beeinflussung der Geräte durch das Energiemanagement sollte aus Akzeptanz- und Sicherheitsgründen nur mit Wissen der Nutzerin bzw. des Nutzers stattfinden. Sofern keine automatisierte Steuerungskomponente vorgesehen ist, könnten die Nutzerinnen und Nutzer durch geeignete Interaktionen davon in Kenntnis gesetzt werden, welche Zeiträume für einen optimierten Betrieb der Geräte geeignet sind. Beleuchtung, Staubsauger und Fernsehgeräte sind Beispiele für diese Klasse.

Ähnliche Definitionen finden sich auch in einigen Literaturstellen zu den Algorithmen in Unterabschnitt 2.2.2 (so z. B. in [Cha+19]). Viele Literaturstellen in Unterabschnitt 2.2.2 unterscheiden Geräte jedoch nur in zwei Klassen, beispielsweise in steuerbare und nicht-steuerbare Geräte bzw. in zeitlich verschiebbare Verbraucher und zeitlich nicht-verschiebbare Verbraucher (so z. B. [Cel+18], [Awa+18], [Li+18a]).

#### 4.1.6 Nachbildung der Funktionalität zur Beeinflussung des Betriebsstatus

Wie in Unterabschnitt 4.1.1 beschrieben, müssen nach dem notwendigen Funktionsumfang zwei Voraussetzungen erfüllt sein, um mittels der automatischen Optimierung Geräte beeinflussen zu können. Zum einen bedarf es einer Schaltfunktion, um den Betriebsstatus eines Geräts zu ändern. Einige Geräte, die über Kompatibilitätsschichten eingebunden werden, verfügen bereits über solche Funktionen. Anderenfalls ist eine Nachrüstung mittels einer schaltbaren Zwischensteckdose möglich. Jedoch ist dadurch nur ein Zu- oder Abschalten des kompletten Geräts möglich.

Bei der zweiten Voraussetzung handelt es sich um eine Funktion, die Prognosen des Energieverbrauchs und möglicher Folgekosten bereitstellt. Diese Funktion soll ebenfalls auf Grundlage der Gerätemodellierung nachgebildet werden. Dazu soll mit Hilfe der Energieverbrauchsermittlung (siehe Unterabschnitt 4.1.4) auf Basis prognostizierter interner Zustände eine Energieverbrauchsprognose vorgenommen werden. Die Prognose der internen Zustände basiert dabei auf dem typischen Verhalten sowie den beabsichtigten Beeinflussungen des Gerätes. Die in Unterabschnitt 4.1.1 benannten Folgekosten können als weitere Ausgabe des

Gerätemodells betrachtet und dem Modell manuell oder per Trainingsalgorithmus angefügt werden. Auf diese Weise können Folgekosten (pro Zeiteinheit) direkt einzelnen internen Zuständen zugeordnet werden.

## 4.2 Einbindung von nicht-vernetzten Geräten

Neben vernetzten Geräten sollen auch nicht-vernetzte Geräte eingebunden werden können. Dies soll aufbauend auf dem Konzept der digitalen *IoT-Gerätetrepräsentation* realisiert werden, bei dem die relevanten Daten und Funktionen auf einem übergeordneten vernetzten Gerät nachgebildet werden. In Unterabschnitt 4.2.1 wird dazu zunächst die Nachbildung des Funktionsumfangs von vernetzten Geräten (siehe Unterabschnitt 4.1.1) durch die IoT-Gerätetrepräsentation vorgestellt.

Dabei wird die Gerätemodellierung aus Unterabschnitt 4.1.3 zugrunde gelegt. Wie ebendort definiert, wird das Verhalten eines Gerätes durch dessen Betriebsstatus abgebildet. In diesem Abschnitt wird die Nachbildung des Betriebsstatus beschrieben, da dieser bei nicht-vernetzten Geräten per Definition nicht zur Verfügung steht. Entscheidend hierbei ist die Qualität der Nachbildung. Wird durch die Nachbildung das Verhalten des Gerätes nur ungenügend abgebildet, können dadurch gegebenenfalls Folgefehler in den Energiemanagement-Anwendungen auftreten. Nachfolgend werden verschiedene Verfahren für die Nachbildung vorgestellt. In der Regel können konkrete Geräte dabei mit mehreren dieser Verfahren eingebunden werden. Die Verfahren unterscheiden sich in ihrer Komplexität und ihrer Einsatzfähigkeit. Sie stellen eine Abwägung zwischen den zusätzlichen Kosten für die Nachbildung und der Qualität der Abbildung dar. Kosten können beispielsweise durch zusätzliche Sensoren auftreten. Dieses Kosten-Nutzen-Verhältnis ist zudem abhängig von der Komplexität des Gerätes.

Die Abbildung 4.2 zeigt eine Übersicht der in diesem Abschnitt betrachteten Verfahren. Zunächst wird nach dem Verhalten der Geräte unterschieden.

- Permanent betriebene Geräte, wie zum Beispiel ungesteuerte, unregelmäßige Heizungs-pumpen, können recht einfach durch feste Werte (Dauer-An) nachgebildet werden. Bei Geräten, die ein festes zeitliches Verhalten besitzen, wie zum Beispiel Anlagen der Gebäudetechnik, kann die Nachbildung mit einfachen mathematischen Mitteln erfolgen (An zwischen 09 und 18 Uhr; Sonst Aus). Für diese Modellierung können **allgemein verfügbare Daten** herangezogen werden, wie z. B. Uhrzeiten, Temperaturen, Intensität der Sonneneinstrahlung und weitere. Dabei können unvorhergesehene Abweichungen im Verhalten der Geräte oder Defekte ohne zusätzliche Informationen bzw. Anbindungen nicht berücksichtigt werden. In solchen Fällen kommt es zu hohen

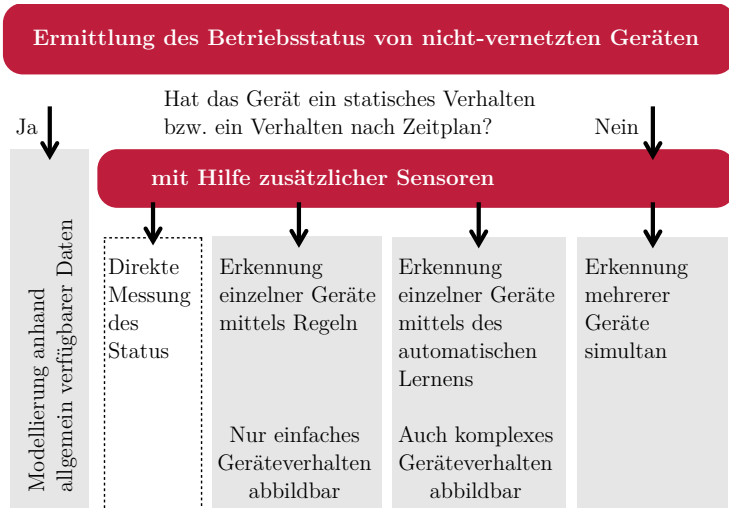


Abbildung 4.2: Übersicht – Wie kann der Betriebsstatus eines Gerätes ermittelt werden?

Fehlern der Nachbildung. Weiter ist zu beachten, dass Geräte, die kein statisches oder zeitlich exakt wiederkehrendes Verhalten aufweisen, durch diese Technik nur mit großen Abweichungen nachgebildet werden.

Durch die Verarbeitung von zusätzlichen Sensordaten – wie auch in Abbildung 4.2 dargestellt – ergeben sich weitere Varianten, die im Folgenden erläutert werden. In Unterabschnitt 4.2.2 wird identifiziert, welche Sensoren genutzt werden könnten. Die Sensordaten können auch direkt von Geräten stammen, die Messdaten über eine vorhandene Kommunikationstechnik liefern, aber keinen nutzbaren Wert für den Betriebsstatus ausgeben. Weiterhin kann die Visualisierung dieser Sensordaten als Teil der IoT-Geräterepräsentation auch den erfahrenen Nutzerinnen und Nutzern bei ihren Analysen dienen.

- Wie Abbildung 4.2 darstellt, kann unter Umständen durch die zusätzlichen Sensoren eine **direkte Messung des Status** erreicht werden. Dies ist beispielsweise der Fall, wenn die Stellung eines Schalters durch eine Spannungsmessung bestimmt wird. Auf diese Variante wird in diesem Kapitel nicht weiter eingegangen.
- In allen anderen Fällen muss der Betriebsstatus durch eine Schätzung aus den Sensordaten ermittelt werden. In Unterabschnitt 4.2.3 wird dazu ein Verfahren vorgestellt, welches aus den zusätzlichen Sensordaten **mit regelbasierten Verfahren** den Betriebsstatus eines Gerätes identifiziert. Diese regelbasierten Verfahren müssen durch

Expertinnen und Experten oder erfahrene Anwenderinnen und Anwender selbst konfiguriert werden.

- In Unterabschnitt 4.2.4 wird diese Technik modifiziert, damit **statistische Trainingsverfahren** angewendet werden können. Hier können Algorithmen des automatisierten Lernens wie Hidden-Markoff-Modelle oder künstliche neuronale Netze eingesetzt werden.
- Zusätzliche Sensoren erhöhen die Kosten für die Anbindung der Geräte. Deshalb wird in Unterabschnitt 4.2.5 abschließend eine Technik beschrieben, die das Ziel verfolgt den Betriebsstatus **mehrerer Geräte** gleichzeitig mit wenigen Sensoren oder sogar nur mit einem Sensor zu erkennen.

Wenn jedes Gerät einer Umgebung mit einem oder sogar mehreren zusätzlichen Sensoren erfasst wird, lässt sich eine theoretisch qualitativ eine bestmögliche Nachbildung erreichen. In der Praxis ist dieses Vorhaben jedoch kostenaufwendig und damit in den meisten Fällen weder sinnvoll noch wirtschaftlich darstellbar. Um möglichst kostengünstig eine ausreichende Abbildungsqualität sicherzustellen, ist deshalb grundsätzlich zu klären, welches Gerät mit welchem Verfahren eingebunden werden kann.

Die Parameter für die Zustandsidentifikation eines Gerätes werden ebenfalls als Modell zusammengefasst. Der Unterabschnitt 4.2.6 befasst sich abschließend mit der Fragestellung, wie diese Modelle vor Ort bereitgestellt werden können.

### 4.2.1 Einbindung als IoT-Gerätetrepräsentation

Bei einer digitalen IoT-Gerätetrepräsentation handelt es sich um einen Datensatz, der ein reales Gerät repräsentiert. Dieser Datensatz ist in einer Instanz des Energiemanagementsystems gespeichert. Die notwendigen Daten und Funktionen zur Interaktion mit den Energiemanagement-Anwendungen und anderen Geräten werden in diesem Datensatz gehalten bzw. im Kontext dieses Datensatzes ausgeführt. Eine Kommunikationsverbindung zu dem realen Gerät ist nicht notwendig, solange sich keine Abweichungen zwischen dem Betriebsstatus des realen Gerätes oder der Repräsentation ergeben.

Zur Betrachtung der notwendigen Eigenschaften der IoT-Gerätetrepräsentationen wird der in Unterabschnitt 4.1.1 beschriebene Funktionsumfang von IoT-Geräten herangezogen. Nachfolgend wird die Umsetzung für die IoT-Gerätetrepräsentation aufgezeigt:

- Damit ein Energiemanagementsystem von der **Existenz** eines einzubindenden Gerätes erfahren kann, muss eine initiale Kommunikation stattfinden. Dies soll bei nicht-

vernetzten Geräten – sofern nicht informationstechnisch möglich – von der Systemeinrichterin bzw. vom Systemeinrichter übernommen werden. Diese bzw. dieser muss dazu das Gerät manuell in das übergeordnete System eintragen. Hierbei muss auch erfasst werden, um welche Geräteart es sich handelt. Dies kann beispielsweise durch die Eingabe der Fabrikat- und Produktbezeichnung des Geräts erfolgen. Daraufhin legt das übergeordnete System eine passende IoT-Geräterepräsentation an und gibt sie dem restlichen Energiemanagementsystem bekannt. Zusätzlich muss die Systemeinrichterin bzw. der Systemeinrichter ggfs. weitere beschreibenden **Metadaten** eingeben, mit denen die Repräsentation angereichert wird. Unterstützungsfunktionen, wie z. B. durch Scannen eines Produktbarcodes sind möglich, aber nicht Teil dieser Arbeit.

- Wie bereits erläutert, muss der **Betriebsstatus** durch die initialisierte IoT-Geräterepräsentation nachgebildet werden. Wie in der Kapiteleinführung erläutert, sollen hierzu verschiedene Techniken eingesetzt werden können. Für die Entwicklung der Nachbildung sollen die unbeschränkten Möglichkeiten der verwendeten Programmiersprache genutzt werden können. In den nachfolgenden Unterabschnitten werden einige geeignete Nachbildungsmethoden beschrieben.
- Ist der Betriebsstatus bekannt, können daraus auch Informationen und fehlende **Sensordaten** nachgebildet werden. Hierzu sei auf die Ansätze aus Unterabschnitt 4.1.4 verwiesen.
- Die Nachbildungen der **optionalen Funktionalität zur Beeinflussung** können bei Bedarf, wie in Unterabschnitt 4.1.6 beschrieben, angewendet werden. Hierzu wird ein zusätzlicher Aktor, wie z. B. eine schaltbare Zwischensteckdose benötigt.
- Die in Unterabschnitt 4.1.1 geforderte **Erweiterungsfähigkeit** muss ebenfalls durch die Geräterepräsentation implementiert werden. Die mögliche Erweiterungsfähigkeit in Hinblick auf das Gerät ist jedoch auf die vorliegenden Daten und Funktionen der IoT-Geräterepräsentation beschränkt.

Die IoT-Geräterepräsentation und ihre Verfahren können auch dazu genutzt werden, vernetzte Geräte in ihrem Verhalten zu überwachen. Weichen die Aussagen von IoT-Gerät und IoT-Repräsentation im Betrieb voneinander ab, könnte dies auf einen Defekt des Gerätes hinweisen.

## 4.2.2 Einbindung mit Hilfe zusätzlicher Sensoren

Trotz der unbeschränkten Implementierungsmöglichkeiten der Nachbildung, können durch die begrenzt zur Verfügung stehenden Daten ggfs. nicht alle Details korrekt nachgebildet



werden. Um die Qualität der Repräsentation weiter zu verbessern, können zusätzliche Sensoren am Gerät oder in der Nähe installiert und die anfallenden Daten zur Ermittlung und Nachbildung herangezogen werden. Von Interesse sind dabei alle Ein- und Ausgaben, wie sie für die Gerätemodellierung in Unterabschnitt 4.1.3 definiert wurden.

Im Folgenden wird erörtert, welche Sensoren für die Anreicherung der Nachbildung prinzipiell geeignet sind. Hierzu werden zunächst einfachere Verknüpfungen herangezogen, wie beispielsweise die Erfassung der Beleuchtung durch ihren Lichtschein. Anschließend werden weitere komplexere Verknüpfungen dargestellt.

- Durch die Messung der **elektrischen Spannung** kann der Schaltzustand von Schaltern, Schützen, Relais usw. detektiert werden. Dadurch kann, wie im vorherigen Unterabschnitt erläutert, beispielsweise auf den Betriebsstatus der angeschlossenen Geräte, z. B. einer Beleuchtung, geschlossen werden.
- Geräte, die elektrische Energie in thermische Energie umsetzen, können ggfs. durch zusätzliche **Temperatursensoren** ergänzt werden. Beispielsweise könnte von der Höhe der Vorlauftemperatur einer Heizung auf den Energieeinsatz der Gesamteinheit geschlossen werden. Auch bei Geräten, die thermische Energie nur als Nebenprodukt erzeugen, kann die Messung von Temperaturen hilfreich sein. Bei der späteren Nutzung der Daten ist jedoch zu beachten, dass die Umgebungstemperatur einen zusätzlichen Einfluss besitzt.
- Durch **Helligkeitssensoren** könnte auf den Status der Beleuchtung geschlossen werden. Dabei ist der Einfluss von anderen Lichtquellen zu berücksichtigen. Weiterhin können durch Helligkeitssensoren auch Statusleuchten an Geräten adaptiert werden.
- Geräte mit bewegten Elementen können mit zusätzlichen **Vibrationssensoren** ausgestattet werden. Auch Näherungssensoren, Lage- bzw. Positionssensoren sowie Beschleunigungssensoren an bewegten Teilen können von Interesse sein. Mögliche Einsatzorte dieser Sensoren sind insbesondere Industriemaschinen.
- Viele Geräte mit bewegten Teilen emittieren charakteristische Schallereignisse bei Aktivierung oder in ihrem Betrieb. So lassen sich die typischen Geräusche bei der Betätigung der Einschaltmechanik eines Toasters oder auch die verschiedenen Schleuderphasen einer Waschmaschine mit einer Geräuscherkennung zuordnen. Auch einige elektronische Geräte, wie Mikrowellen, emittieren wahrnehmbare akustische Signale. Entsprechend eignen sich auch **Mikrofone** zur Aufnahme der Aktivität von Geräten. Jedoch nehmen Mikrofone alle Schallereignisse der Umgebung auf, darunter auch Schallereignisse anderer Geräte sowie Fremdgeräusche. Die Aufzeichnungen sind also nicht eindeutig direkt einem einzelnen Gerät zuzuordnen.

- Bewegliche Teile und Statusleuchten können auch durch **Kameras** aufgenommen und durch eine Bildverarbeitung detektiert werden. Kameras sind in der Lage auch aus Distanz Objekte wahrzunehmen, sofern eine Sichtverbindung gegeben ist. Eine solche Verarbeitung ist aufwändig, kann insbesondere im industriellen Kontext aber sinnvoll sein.
- Auch über den Zugriff auf Sensordaten, die als Ist-Werte in eine Regelung einfließen, kann auf den Betriebszustand geschlossen werden. Beispiele hierfür sind neben den bereits benannten Temperatursensoren auch **Sensoren zur Luftqualität** (Luftfeuchte, CO<sub>2</sub>-Gehalt, usw.)
- Zudem ist denkbar, dass mittels Magnetfeldsensoren oder der Detektion von elektromagnetischer Strahlung Ableitungen zum Betriebszustand spezifischer Geräte möglich sein könnten.
- Verbrauchsmessgeräte, z. B. für Wasser, Druckluft, Wärme, Kälte sowie für elektrische Energie, können ebenfalls geeignete Messreihen zur Zuordnung des Betriebszustands darstellen. Da es sich hier um Abrechnungsgrößen handelt, sind diese auch für die direkte Verwendung beispielsweise bei der Visualisierung für Expertinnen und Experten oder für das Lastspitzenmanagement interessant. Jedoch sind Verbrauchsmessgeräte im Vergleich zu vielen anderen Sensoren sowohl in der Anschaffung als auch im Betrieb kostenaufwendiger.

Da die Ein- bzw. Ausgaben, wie in Unterabschnitt 4.1.3 erläutert, in Abhängigkeit zum Betriebsstatus stehen, ist es ggfs. möglich, diesen aus den Messdaten zu identifizieren. Dieses Prinzip wird nachfolgend als Zustandsidentifikation bezeichnet. Aufbauend auf den hier beschriebenen Sensoren werden in den nächsten Unterabschnitten verschiedene Verfahren beschrieben.

### 4.2.3 Zustandsidentifikation mittels regelbasierter Verfahren

Eine Zustandsidentifikation kann durch individuelle Softwaremodule und durch die folgenden Standardverfahren abgebildet werden. In der späteren Implementierung sollen diese Verfahren ohne weitere Implementierungsarbeiten nur durch eine Parametrisierung verwendet werden können. Die Standardverfahren bauen auf den Verfahren der Signalverarbeitung auf. Zusätzlich wird die in Unterabschnitt 4.1.3 vorgestellte Gerätemodellierung angewendet. Im Unterschied zu Unterabschnitt 4.1.3 werden hier die internen Zustände eines Geräts nur als

eindimensionaler, diskreter Betriebsstatus

$$s \in \mathbb{S}^s = \{S_1, \dots, S_N\}$$

mit  $N$  Zuständen modelliert. Das aktuelle Verhalten eines Geräts wird vollständig durch diesen Betriebsstatus charakterisiert. Der Betriebsstatus ist ausschließlich abhängig von den Eingaben (siehe Unterabschnitt 4.1.3). So z. B. Eingaben durch Nutzerinnen oder Nutzer, die zu einer Veränderung des Betriebsmodus führen, von Umweltdaten oder der Tageszeit. Die beobachtbaren Ausgaben sind wiederum abhängig vom Betriebsstatus. Sind nun alle Eingaben eines Gerätes bekannt, so kann – wie bereits in der Einleitung zu Abschnitt 4.2 erläutert – direkt auf den Betriebsstatus geschlossen werden, da es sich hierbei um eine direkte Ursache-Wirkung-Beziehung handelt. Verfahren, die von der Ursache auf die Wirkung schließen, wurden bereits in Unterabschnitt 4.1.4 erläutert und können für die Ableitung des Betriebsstatus aus Eingangsgrößen adaptiert werden.

Unter Betrachtung der in Unterabschnitt 4.2.2 aufgeführten Sensoren, wird deutlich, dass es sich bis auf den direkten Abgriff der Schaltbefehle sowie bei den Sensordaten in der Regel um Ausgaben nach der Gerätemodellierung handelt. In diesem Abschnitt soll deshalb dargestellt werden, wie von Ausgaben eines Gerätes auf den Betriebsstatus geschlossen werden kann. Hierbei muss also von der Wirkung auf die Ursache zurück geschlossen werden. Da verschiedene Ursachen die gleiche Wirkung auslösen können, ist diese Abbildung nicht immer zweifelsfrei möglich.

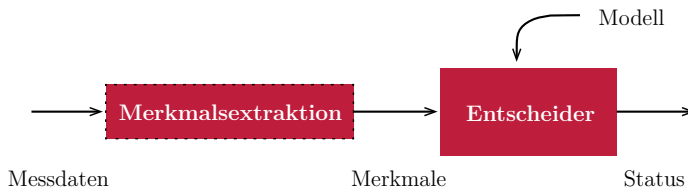


Abbildung 4.3: (Regelbasiertes) Verarbeitungsschema zur Zustandsidentifikation anhand von Messdaten. Eine optionale Merkmalsextraktion erlaubt eine variable Vorverarbeitung der Messdaten. Die Parameter des Entscheiders werden als Modell zusammengefasst.

Für die Zustandsidentifikation anhand von Ausgangsgrößen wird die in Abbildung 4.3 dargestellte Grundstruktur vorgeschlagen. Dieses Verfahren besteht aus zwei Elementen: Einer optionalen Transformation der Sensordaten und einem regelbasierten Entscheider.

Der regelbasierte Entscheider stellt das Kernelement dar. Dabei wird die Zustandsausprägung

$S_j \in \mathbb{S}^*$  als geschätzter Zustand  $\hat{s}(t)$  zum diskreten Zeitpunkt  $t$ , mit

$$\hat{s}(t) = \begin{cases} S_j, & \text{wenn } u_j < x(t) < o_j, \quad \forall j \in \{1, \dots, J\} \\ s(t-1), & \text{sonst} \end{cases},$$

übernommen, wenn der Eingangswert des Entscheiders  $x(t)$  innerhalb der unteren und oberen Schranke  $u_j$  und  $o_j$  mit  $u_j, o_j \in \mathbb{R}$  liegt. Diese Bedingung wird für alle  $J$  Intervalle geprüft. Liegt der Eingangswert in mehreren Bereichen, wird die erste Übereinstimmung ausgewählt. Sollte der Eingangswert in keinem Intervall liegen, so wird kein neuer Zustand ausgewählt, sondern der Zustand  $s(t-1)$  des vorherigen Zeitpunkts  $t-1$  übernommen. Der Eingangswert  $x(t)$  und damit auch die unteren und oberen Schwellen  $u_j$  und  $o_j$  sollen dabei auch als Tupel  $\vec{u}_j$  und  $\vec{o}_j$ , also als eine Kombination mehrerer Eingangssignale, auftreten dürfen. Dadurch ergibt sich die mathematische Formulierung

$$\hat{s}(t) = \begin{cases} S_j, & \text{wenn } \vec{u}_j < \vec{x}(t) < \vec{o}_j, \quad \forall j \in \{1, \dots, J\} \\ s(t-1), & \text{sonst} \end{cases}.$$

Die Kombination aus allen Entscheiderschwellen und ihren zugeordneten Zuständen wird folgend als Regelsatz

$$\lambda = \{\vec{o}_j, \vec{u}_j, S_j | j \in \{1, \dots, J\}\}$$

bezeichnet.

Um den Entscheider flexibler auf Signalkomponenten reagieren lassen zu können, soll optional eine Transformation  $T_e$ , mit

$$\vec{x}(t) = T_e(\vec{e}(t)),$$

zwischen den Sensordaten  $\vec{e}(t)$  und dem Entscheider eingefügt werden können. Die Sensordaten, hier bereits als Tupel dargestellt, können sowohl Daten eines einzelnen Sensors als auch mehrerer Sensoren zu einem Zeitpunkt darstellen. Die Transformation soll nicht-lineare Operationen enthalten dürfen. Im Sinne einer *Merkmalsextraktion* (siehe [PK08, S. 294f]) soll diese Transformation Informationen aus dem Signal extrahieren können, anhand deren Entscheidungen getroffen werden können. Dabei kann es sich zum Beispiel um eine Transformation in den Frequenzbereich handeln. Soll keine Transformation zum Einsatz kommen, können die Sensordaten direkt als Eingangswerte für den Erkenner übernommen werden.

Zu beachten ist, dass sich die Veränderung der Transformation auch auf die Entscheiderschwellen auswirkt. Entsprechend müssen die Transformation und die Regelsätze aufeinander angepasst sein. Deshalb wird die Kombination aus Transformation und den Regelsätzen im Folgenden als Modell  $\mathcal{M} = \{T_e, \lambda\}$  bezeichnet.

#### 4.2.4 Zustandsidentifikation mittels statistischer Verfahren

Die manuelle Erstellung von Modellen ist aufwendig und bedarf besonderen Expertenwissens. Aus diesem Grund wird nachfolgend erläutert, wie unter Zuhilfenahme statistischer Schätzverfahren Modelle automatisch gewonnen werden können.

Dazu wird auf dem in Unterabschnitt 4.2.3 erläuterten Verfahren zur Erkennung aufgesetzt. An Stelle der regelbasierten Entscheidung wird nun auf Entscheidungen anhand von Wahrscheinlichkeiten gesetzt. Dazu wird ein *Maximum a-posteriori* (MAP) Entscheider (vgl. [PK08, S. 120]) verwendet, der mit

$$\hat{s}(t) = \arg \max_s (P(s|s(t-1), s(t-2), \dots, \vec{x}(t), \vec{x}(t-1), \dots))$$

jeweils den Zustand  $\hat{s}$  auswählt, der in diesem Moment  $t$  die höchste Wahrscheinlichkeit  $P(s|\dots)$  aufweist. Der Term  $s(t-1), s(t-2), \dots, \vec{x}(t), \vec{x}(t-1), \dots$  stellt dabei die Größen dar, von denen der aktuelle Zustand abhängig ist. Abgeleitet aus dem Gerätemodell (siehe Unterabschnitt 4.1.3) wurden hier die vorherigen Zustände  $s(t-1), s(t-2), \dots$  und die aktuellen sowie die vorherigen Beobachtungen  $\vec{x}(t), \vec{x}(t-1), \dots$  eingetragen. Für die Berechnung der Wahrscheinlichkeit eines Zustands (oder eine Abfolge von mehreren Zuständen) kommen verschiedene Schätz- und Trainingsverfahren in Frage. Nachfolgend werden exemplarisch zwei Verfahren vorgestellt und adaptiert. Die Verfahren setzen verschiedene weitere Vereinfachungen in Bezug auf die Abhängigkeiten voraus:

*Hidden-Markoff-Modelle* (engl. Hidden Markov Model, HMM) eignen sich dazu, aus beobachtbaren verursachenden Größen auf verdeckte, also nicht direkt messbare Zustände zurückzuschließen (siehe [Rab89]). Damit sind sie für die Geräteerkennung prädestiniert, bei der von verursachenden Größen auf die verdeckten Betriebsstatus geschlossen werden soll. Hidden-Markoff-Modelle erster Ordnung bedingen, dass die oben genannte Wahrscheinlichkeit ausschließlich von dem vorherigen Zustand  $s(t-1)$  und der aktuellen Beobachtung  $\vec{x}(t)$  abhängig ist. Entsprechend vereinfacht sich der Entscheider zu

$$\hat{s}_{\text{HMM}}(t) = \arg \max_s (P(s|s(t-1), \vec{x}(t))),$$

da die weiter zurückliegenden Zustände und Beobachtungen entfallen. Es wird davon ausgegangen, dass deren Informationen bereits im Zustand  $s(t-1)$  enthalten sind.

*Künstliche neuronale Netze* (engl. Artificial Neural Network, ANN) setzen im Gegensatz zu Hidden-Markoff-Modellen nicht auf eine Berechnung der exakten Wahrscheinlichkeit, sondern ermitteln eine möglichst genaue Reproduktion eines Wertes oder einer Sequenz anhand der Eingangswerte (siehe [PK08, S. 415ff]). Je nach ihrer Netzstruktur können

sie dabei verschiedene Abhängigkeiten beachten. Folgende drei Beispiele modellieren die Abhängigkeit 1.) nur zu den aktuellen Eingangsgrößen, 2.) zu den aktuellen und vergangenen Eingangsgrößen, 3.) zu den Eingangsgrößen und dem vorangegangenen Zustand.

$$\hat{s}_{\text{ANN1}}(t) = T_{\text{ANN1}}(\vec{x}(t))$$

$$\hat{s}_{\text{ANN2}}(t) = T_{\text{ANN2}}(\vec{x}(t), \vec{x}(t-1))$$

$$\hat{s}_{\text{ANN3}}(t) = T_{\text{ANN3}}(\vec{x}(t), \dots, s_{\text{ANN3}}(t-1))$$

Die Merkmalsextraktion, die aus Unterabschnitt 4.2.3 unverändert übernommen wird, muss weiterhin von Expertinnen und Experten ausgewählt und parametrisiert werden.

Für die Erstellung der Modelle, auch Training genannt, werden Datenbanken bestehend aus Messdaten und den Sequenzen der tatsächlich angenommenen Zustände benötigt. Die Zustände werden nachfolgend auch als Wahrheiten bezeichnet. Eine solche Datenbank wird für das Training in zwei Teile geteilt: Einen Trainings-Bereich und einen Test-Bereich. Der Trainings-Bereich dient zur Berechnung der Parameter des Schätzers. Mit Hilfe des Test-Bereichs kann anschließend evaluiert werden, wie gut der Schätzer die tatsächlichen Zustände erkennt.

#### 4.2.5 Zustandsidentifikation mehrerer Geräte

Die Installation zusätzlicher Sensoren, insbesondere auch für die Erfassung des elektrischen Energieverbrauchs, führt zu zusätzlichen Kosten in der Anschaffung und im Betrieb. Ggfs. führt dies dazu, dass das gesamte System nicht wirtschaftlich zu betreiben ist. Wünschenswert ist also eine Minimierung der Anzahl der verwendeten Sensoren und gleichzeitig die Möglichkeit, den Betriebsstatus gleich mehrerer Geräte über nur einen Sensor zu erkennen. Diese Technik wird dem Themenfeld der *Disaggregation* zugeordnet. Sie wird im Englischen auch als *Nonintrusive Load Monitoring* (engl., NILM) bezeichnet. Dieses Themenfeld wurde bereits in Unterabschnitt 2.2.3 vorgestellt.

Bei dieser Art der Erkennung muss beachtet werden, dass verschiedene Geräte gleiche Wirkungen auf die Sensordaten zeigen können. Bereits hier ist offensichtlich, dass zwei identische Geräte, sofern sie keine weiteren Unterscheidungskriterien besitzen, nicht voneinander zu unterscheiden sind. Auch Geräte, die sich in ihrem Verhalten oder ihren beobachtbaren Ausgaben ähneln, sind wahrscheinlich schwierig voneinander zu unterscheiden.

Deshalb sollte bereits bei der Auswahl der Sensoren darauf geachtet werden, dass diese zu den Geräten und ihren Ausgaben passen. Um Geräte gut voneinander zu unterscheiden, sind differenzierbare Ausprägungen in den Messdaten erforderlich. Dabei können insbesondere die folgenden Orthogonalitäten ausgenutzt werden (motiviert aus [Yan16]):

- **Ereignisse** sind charakteristische Signalsequenzen, die nur für kurze Zeit auftreten und sich dadurch in der Regel nicht überlappen. Sie stehen für eine Veränderung, beispielsweise eines Betriebszustands. Wird zum Beispiel die Gesamtleistungsaufnahme mehrerer Geräte gemessen, so wird das Einschalten eines einzelnen Verbrauchers, z. B. eines Beleuchtungskörpers anhand eines für diesen Verbraucher vorher ermittelten und damit bekannten Leistungsanstiegs deutlich. Unter Betrachtung von Unterabschnitt 4.2.2 kommen hier Verbrauchs-, Audio-, Video- und Vibrationssensoren in Frage.
- **Merkmale** stellen dauerhafte oder wiederkehrende charakteristische Signalanteile dar. Sie können z. B. mit einer schnellen Fouriertransformation separiert werden. Von den Sensoren aus Unterabschnitt 4.2.2 kommen mit Ausnahme der Sensoren für die Schaltzustände alle dort aufgeführten Sensoren in Frage. Jedoch kann vermutet werden, dass sich Helligkeits-, Temperatur-, Luftfeuchte-, Luftqualitätssensoren nur bedingt eignen, da nur wenige Geräte entsprechende Ausgaben emittieren. Weil alle Geräte für ihren Betrieb Energie benötigen, sind insbesondere Verbrauchssensoren von starkem Interesse. Aber auch Mikrofone, Kameras und elektromagnetische Sensoren sind aufgrund ihrer vielseitigen Einsatzmöglichkeiten und ihres hohen Auflösungsvermögens interessant.

Auch für die Geräteerkennung wird die Grundstruktur für die Zustandsidentifikation aus Unterabschnitt 4.2.3 verwendet und der Erkenner angepasst. Anstelle eines Betriebsstatus muss nun das Tupel der Betriebszustände  $\vec{s}(t) = [s_1(t), \dots, s_k(t), \dots, s_K(t)]'$  von  $K$  Geräten erkannt werden. Der Zustand eines Einzelgerätes  $s_k(t) \in \mathbb{S}_k$  mit individuellem Zustandsraum  $\mathbb{S}_k$  wird dabei mittels

$$\hat{s}_k(t) = \arg \max_s (P(s_k | \vec{s}(t-1), \vec{s}(t-2), \dots, \vec{x}(t), \vec{x}(t-1), \dots)), \quad \forall k \in \{1, \dots, K\})$$

geschätzt. Hierbei wurden im Vergleich zur Einzelgeräteerkennung in Unterabschnitt 4.2.4 auch die vergangenen Zustände aller anderen Geräte  $s_l(t-\tau)$  mit  $\tau > 0$  und  $l \in \{1, \dots, K\} \setminus \{k\}$  als Abhängigkeiten hinzugenommen.

Aufgrund der gleichen Grundstruktur können die Konzeptionen aus Unterabschnitt 4.2.3 und 4.2.4 angewendet werden. Bei der Durchführung vieler Trainingsverfahren müssen Messdaten von allen Gerätekombinationen aufgenommen werden, welches bei nur sechs Geräten ( $G$ ) und zwei Ausprägungen ( $A$ ) des Betriebsstatus bereits  $A^G - 1 = 63$  Fälle

ergeben. Wenn die Messdaten der Geräte tatsächlich unabhängig voneinander sind, könnten durch eine geeignete Trainingsstruktur die Kombinationen synthetisch aus den Messdaten der Einzelgeräte erzeugt werden. Dadurch wären nur noch  $G \cdot A = 12$  Fälle zu ermitteln.

#### 4.2.6 Bereitstellung von Modellen

Wie beschrieben, ist für die Erkennung eines Gerätes ein angepasstes Modell erforderlich. Im Folgenden werden die Möglichkeiten zur Bereitstellung von Modellen dargestellt. Dabei sind verschiedene Abhängigkeiten zu berücksichtigen. Ein Modell kann sich auf ganze Geräteklassen oder auch nur auf ein spezielles Produkt beziehen. Neben den Abhängigkeiten innerhalb eines Modells ist auch eine Kompatibilität mit den verwendeten Messdaten, deren Wertskalierung und Abtastraten erforderlich.

Aus Sicht der Anwenderinnen- bzw. Anwender können Modelle

- von einer erfahrenen Nutzerin bzw. von einem erfahrenen Nutzer ermittelt und eingegeben werden,
- für das spezielle Gerät aus dem Internet geladen werden, welches die Ähnlichkeit von Geräten und Messstellen mit Hilfe eines Internet-Datenbanksystems berücksichtigt,
- für die Gerätekategorie aus dem Internet geladen werden und auf das Gerät (automatisch oder manuell) angepasst werden sowie
- lokal speziell für ein Gerät trainiert werden. Dazu bedarf es einer Trainingsphase, in der die tatsächlich angenommenen Zustände bekannt sind.

### 4.3 Kommunikationsstruktur für Energiemanagement-Systeme

Die mittlere Schicht des Schichtenmodells vermittelt zwischen den Geräten bzw. deren Technik zur Einbindung (siehe Abschnitt 4.1 und 4.2) und den Funktionalitäten zum Energiemanagement. In diesem Abschnitt wird eine Datenkommunikation und -verarbeitung auf Basis einer hierarchischen Kommunikationsstruktur vorgestellt, die eine Kommunikation unter Einhaltung von Datenschutz- und Datensicherheitsrichtlinien erlaubt. Für diese Kommunikationsaufgabe müssen folgende Aspekte betrachtet werden:

In Unterabschnitt 4.3.1 wird die Datenkommunikation für eine Protokoll-basierte Datenübertragung beschrieben. Dabei wird kein konkretes Protokoll betrachtet, sondern es werden



die grundsätzliche Funktionsweise sowie die erforderlichen Eigenschaften beschrieben. Die Beschreibung fußt auf dem Funktionsumfang von Geräten aus Unterabschnitt 4.1.1.

Anschließend wird in Unterabschnitt 4.3.2 die bereits benannte hierarchische Kommunikationsstruktur ausgestaltet. Um den Datenaustausch zur Wahrung des Datenschutzes bzw. der Datensicherheit zu begrenzen, wird in Unterabschnitt 4.3.3 ein Konzept basierend auf Regeln vorgestellt. Mit diesen Regeln kann definiert werden, welche Datensätze wohin weitergeleitet werden dürfen.

Die Datenverarbeitung muss ebenfalls lokal in den jeweiligen Umgebungen durchgeführt werden, da nur hier die benötigten Daten zur Verfügung stehen. Entsprechend wird in Unterabschnitt 4.3.4 ein Konzept zum Entwurf von Anwendungen nach einem verteilten Software-Ansatz vorgestellt. Dabei werden einzelne Software-Elemente verteilt auf verschiedene Instanzen in den jeweiligen Umgebungen ausgeführt, die zusammen die gewünschte Anwendung ermöglichen.

Damit eine Ausführung dieser verteilten Anwendungen ermöglicht werden kann, wird in Unterabschnitt 4.3.5 die Verwendung einer einheitlichen Softwareplattform vorgeschlagen.

### 4.3.1 Datenobjekte und deren Austausch

Wie bereits in Abschnitt 4.1 und 4.2 beschrieben, werden von den Geräten sowohl fortlaufende Datenreihen z. B. über den Energieverbrauch bereitgestellt und auch Funktionen z. B. zur Beeinflussung des Betriebsstatus angeboten. Diese werden nachfolgend allgemein als Datenobjekte mit Dateneinträgen und eigenen individuellen Methoden aufgefasst. Um die Eigenschaften der Datenobjekte zu kennzeichnen, werden die Objekte zusätzlich durch Metadaten beschrieben.

Wie in Unterabschnitt 4.1.1 bezüglich der Einbindung von Geräten beschrieben, muss auch hier zunächst dem entfernten System die Existenz der Datenobjekte bekannt gegeben werden. Dazu sollen initial beim Verbindungsaufbau zwischen zwei Systemen jeweils Listen der Datenobjekte ausgetauscht werden. Die Metadaten der Objekte sollen dabei bereits mitübertragen werden. Da sich die Listen der Datenobjekte während der Laufzeit ändern können, beispielsweise wenn ein neues Gerät hinzugefügt wird, sollen diese Listen und deren Änderungen durch ein Publish-Subscribe-Entwurfsmuster (siehe Unterabschnitt 2.3.3) übermittelt werden.

Wie in Anforderung F.2 in Unterabschnitt 3.3.4 benannt, soll die Kommunikation für die anderen Schichten transparent ablaufen. Die obere, wie auch die untere Schicht sollen demnach keine Aufgaben der Datenkommunikation übernehmen. Folglich darf sich auch der

Zugriff auf lokale und entfernte Daten nicht unterscheiden. Für die anderen Schichten darf nicht erkennbar sein, ob ein Datenobjekt lokal oder auf anderen Systemen erfasst und dann erst auf das lokale System übermittelt wurde.

Bei Bedarf muss ein entferntes System einzelne Datenreihen der erhaltenen Datenobjekte abonnieren können. Dazu soll ebenfalls ein Publish-Subscribe-Entwurfsmuster verwendet werden. Dies ermöglicht eine unmittelbare Datenweiterleitung, da neue Einträge einer zuvor abonnierten Datenreihe ohne Wartezeit an das Ziel übermittelt werden können. Anwendung findet hier also der Ansatz der Ereignis-basierten Plattform nach Unterabschnitt 2.3.3.

Die benannten Methoden eines Datenobjekts können nicht an andere Systeme übertragen werden. Sie müssen im Kontext des originalen Datenobjekts ausgeführt werden. Dazu werden diese Methoden im Sinne der Remote-Procedure-Calls (RPC, siehe Unterabschnitt 2.3.4) zum entfernten Aufruf bereitgestellt. Bei einem solchem Aufruf wird die adressierte Methode auf dem initialen System des Datenobjekts ausgeführt und das Resultat des Aufrufs wird zurück an das aufrufende System übermittelt. Anwendung findet hier also zusätzlich der Ansatz der Dienst-orientierten Plattformen nach Unterabschnitt 2.3.3. Dies ist erforderlich, da bei reinen Ereignis-basierten Plattformen zwei Kommunikationspartner nicht direkt bidirektional miteinander kommunizieren können.

### 4.3.2 Hierarchischer Kommunikationsansatz

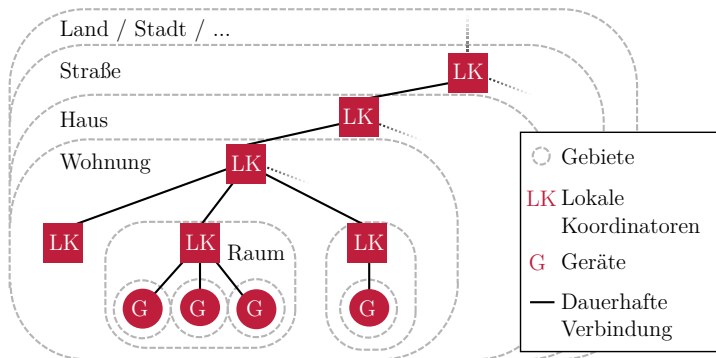


Abbildung 4.4: Durch die Einbringung lokaler Koordinatoren (LK) ergibt sich gemeinsam mit den Geräten (G) eine hierarchische Struktur der Kommunikation. Die einzelnen Umgebungen, in die die lokalen Koordinatoren eingebracht wurden, sind mit gestrichelten Linien angedeutet.

Der Grundgedanke zur Verbesserung des Datenschutzes besteht darin, dass Daten möglichst lokal verarbeitet werden sollen und damit nicht über weite Strecken kommuniziert werden

müssen. Um dieses zu ermöglichen, werden zusätzliche Geräte in die jeweiligen Gebiete eingebracht. Diese verwalten die restlichen Geräte und ermöglichen, dass die Daten der Geräte lokal ausgetauscht werden können. Diese zusätzlichen Geräte werden folgend als *lokale Koordinatoren* (LK) bezeichnet. Alle anderen Geräte (G) sowie andere weiter untergeordnete lokale Koordinatoren eines Gebietes verbinden sich jeweils mit einem (oder auch mehreren) lokalen Koordinatoren. Wie Abbildung 4.4 zeigt, prägt sich dadurch eine hierarchische Kommunikationsstruktur aus. Die Gebiete sind in der Abbildung durch gestrichelte Linien angedeutet. Wie durch die Linien deutlich wird, ergeben sich damit folgerichtig auch Systemgrenzen, an denen Datenschutz und Datensicherheitsanforderungen durchgesetzt werden können.

Die lokalen Koordinatoren verwalten jeweils ihre Gebiete und ermöglichen, dass Daten der Geräte lokal weitergegeben und verarbeitet werden können. Werden Daten von einem höheren lokalen Koordinator benötigt, kann dieser die Daten durch das Publish-Subscribe-Entwurfsmuster anfordern. Damit können bei Bedarf auch weit entfernte Daten über mehrere lokale Koordinatoren hinweg bereitgestellt werden.

Daten dürfen von den lokalen Koordinatoren oder Geräten jedoch nur über die Kommunikationsverbindungen weitergegeben werden und damit die Systemgrenzen verlassen, wenn dieses mit den Datenschutz- und den Datensicherheitsanforderungen vereinbar ist oder explizit von einer Nutzerin oder einem Nutzer erlaubt wurde. Dies gilt auch für die Methoden der Datenobjekte, die vor unberechtigtem Fremdzugriff geschützt werden müssen. Die dazu notwendigen Regeln müssen auf jedem lokalen Koordinator und jedem Gerät separat eingestellt werden.

Dieses Prinzip der hierarchischen Kommunikationsstruktur entspricht dem vermischten Kommunikationsansatz aus Unterabschnitt 2.3.2. Es vereinigt die Systemtypen 0, 1 und 2 aus Unterabschnitt 2.3.5, da sowohl lokale wie auch entfernte Verarbeitungen unter der Verwendung einer offenen Plattform ermöglicht werden. Der Datenschutz kann dabei wie bei Systemtyp 0 gewährleistet werden.

Der Kommunikationsansatz ähnelt der in Unterabschnitt 2.3.5 beschriebenen Gemeinschaftshierarchie, jedoch ohne Festlegung der Anzahl und der Aufgaben der Kommunikationsebenen. Über die Kommunikationsstruktur sollen im Gegensatz zur dort beschriebenen Lösung nicht nur Informationen über die Struktur aggregiert werden, sondern auch Steuerbefehle für die automatisierte Energieverbrauchsoptimierung ausgetauscht werden können.

### 4.3.3 Wahrung des Datenschutzes und der Datensicherheit

Damit die Datensicherheit und der Datenschutz in jedem lokalen Koordinator gewahrt bleiben kann, muss das System selbst entscheiden können, welche Daten an welchen Empfänger weitergeleitet werden dürfen. Wie im vorherigen Unterabschnitt vorgeschlagen, sollen dazu Regeln definiert werden können. Diese legen fest, unter welchen Bedingungen Datenobjekte an Empfänger weitergeleitet werden dürfen. Eine Regel kann für diese Datenobjekte eine der nachfolgenden Aktionen festlegen:

1. Vollständige Weitergabe von Name, Metadaten, Datenreihen und Methoden.
2. Nur Weitergabe von Name, Metadaten und Datenreihen.
3. Wie 2., jedoch sollen die Datenreihen vorverarbeitet werden. Durch ein Resampling der Datenreihen kann die zeitliche Auflösung der Datenreihe verringert werden. Dadurch können ggfs. weniger Informationen aus den Daten abgeleitet werden. Gleiches soll auch durch die optionale Aggregation mehrerer Datenreihen erreicht werden. Anstatt beispielsweise die Datenreihen des Energieverbrauchs aller Einzelgeräte einer Wohnung weiterzugeben, soll nur der aggregierte Verbrauch aller Geräte als eine einzige Datenreihe weitergeleitet werden.
4. Wie 2., jedoch sollen die Datenreihen mit symmetrischen oder unsymmetrischen (Public-Key) Schlüsseln verschlüsselt werden. Dadurch ist eine Datenreihe nur noch für die Empfänger lesbar, die auch die passenden Schlüssel besitzen.
5. Nur Weitergabe des Namens und der Metadaten eines Datenobjekts.
6. Keine Weitergabe des Datenobjekts.

Die Regeln sollen selbstständig vom System, aber auch von der Nutzerin bzw. vom Nutzer vorgegeben werden können.

### 4.3.4 Verteilte IoT-Anwendungen

Die gewünschten Funktionalitäten der oberen Schicht sollen durch eigenständige Anwendungen realisiert werden, die die Funktionalität des Basissystems erweitern. Als erklärendes Beispiel aus einem anderen Bereich sei auf die möglichen Funktionserweiterungen bei Smartphones durch die zusätzliche Installation weiterer „Apps“ hingewiesen. Durch diese Erweiterungsfähigkeit wird auch Drittanbietern ermöglicht, weitere Funktionalitäten in das System einzubringen.

Die Anwendungen arbeiten auf der Grundlage der Datenobjekte der Geräte. Wie beschrieben, sind diese Datenobjekte auf den Geräten oder lokalen Koordinatoren in den jeweiligen Umgebungen verteilt und dürfen diese unter Umständen nicht verlassen (siehe Unterabschnitt 4.3.1). Wenn eine Anwendung für ihre Aufgabe mehrere Datenobjekte benötigt, ist es auch möglich, dass diese auf unterschiedlichen Geräten bzw. lokalen Koordinatoren liegen und diese nicht verlassen dürfen. Die Verarbeitung der Anwendungen kann deshalb nur lokal in den jeweiligen Geräten bzw. lokalen Koordinatoren stattfinden, in denen auch die Datenobjekte vorliegen. Dazu wird im Folgenden ein Softwareentwurf nach einem verteilten Anwendungskonzept vorgeschlagen, der die lokale Verarbeitung von verteilten Daten ermöglichen soll. Die Anwendungen dieses Anwendungskonzepts werden nachfolgend als verteilte IoT-Anwendungen bezeichnet. Das Konzept entspricht dem Ansatz der sicheren Ausführung von Anwendungen als virtuelle Maschinen (VM) aus Unterabschnitt 2.3.3. Für das Systemkonzept kommen nun also neben den Ansätzen von Ereignis-basierten und Dienst-orientierten Plattformen auch die Ansätze von VM-basierten Plattformen zum Einsatz.

Für die Entwicklung soll die folgende Softwarekonzeption zugrunde gelegt werden: Die Funktionalitäten (nun auch IoT-Anwendungen genannt) werden hierzu in sogenannte Software-Elemente unterteilt. Diese Software-Elemente werden auf den beteiligten Geräten ausgeführt. In Analogie zur Kommunikation der Geräte in Unterabschnitt 4.3.1 sollen auch die Software-Elemente Datenobjekte (siehe Unterabschnitt 4.3.1) erstellen und austauschen können, die die Daten und Methoden der einzelnen Software-Elemente beschreiben. Mit Hilfe dieser Datenobjekte sollen die Software-Elemente einer IoT-Anwendung miteinander interagieren.

Übernehmen Geräte aus Sicht der IoT-Anwendung die gleiche Aufgabe, sollen auch die gleichen Software-Elemente der IoT-Anwendung auf diesen Geräten ausgeführt werden. Diese Geräte bilden damit eine sogenannte Geräte- oder auch Funktionsgruppe. Da sich verschiedene Anwendungen unterscheiden, unterscheidet sich in der Regel auch die Aufteilung der Funktionsgruppen und damit auch die Unterteilung der Software-Elemente. Bei der Realisierung eines Anwendungsfalls soll keine Gerätegruppe ausgespart werden, damit keine separaten Anwendungen abseits der IoT-Anwendung realisiert werden müssen. So sollen Software-Elemente von IoT-Anwendungen auch auf Geräten zur Nutzerinteraktion, wie Smartphones, Tablets usw. installiert und ausgeführt werden können. Durch diese Inklusion aller Gerätegruppen wäre es folglich möglich, eine Funktionalität durchweg in einer einzigen Anwendung und damit durch eine einzige Konzeption zu realisieren. Häufig werden in der Praxis für diese Gerätegruppen separate Anwendungen implementiert, welches zu einem erheblichen zusätzlichen Aufwand führt (siehe Unterabschnitt 2.3.2).

Sofern erforderlich, soll eine IoT-Anwendung auch eine Nutzerschnittstelle anbieten können.

Dazu sollen die Software-Elemente in diesen Gerätegruppen Bildschirmanichten definieren und der Nutzerin bzw. dem Nutzer anzeigen können.

Bei einigen Anwendungsfällen müssen einheitliche Entscheidungen getroffen werden, die von mehreren Geräten umgesetzt werden sollen. Um eine Koordination dieser Entscheidungen zu ermöglichen, kann ein separates Gerät eingebracht oder ein bestehendes Gerät mitbenutzt werden. Die Funktionsgruppe dieser Aufgabe wird nachfolgend als Koordination bezeichnet und besteht in einer Umgebung, sofern nicht redundant vorgehalten, in der Regel nur aus einem Gerät. Viele Anwendungsfälle bedingen, dass dieses Gerät ununterbrochen zur Verfügung steht und nicht zeitweise ausgeschaltet wird. Mit dieser Aufgabe sollten in der Regel die eingebrachten lokalen Koordinatoren betraut werden.

Mit dieser Gerätegruppe erhält dieses System lokal wieder einen zentralen Charakter. Fällt das Gerät zur Koordination aus, führen die verbleibenden Geräte jedoch weiterhin Software-Elemente aus. Der sonst bei zentralen Strukturen unvermeidliche Systemausfall bleibt aus. Die einzelnen Software-Elemente können zusätzlich Programmroutinen enthalten und dadurch auf diesen Fall reagieren. Dadurch ist auch ein autonomer Betrieb der einzelnen Geräte möglich, welches eine hohe Ausfallsicherheit trotz der potentiell hohen Anzahl der beteiligten Geräte ermöglicht.

Das Konzept der verteilten IoT-Anwendungen ähnelt dem Konzept der verteilten Anwendungen nach Tanenbaum (siehe [ST17, S. 2]). Bei Tanenbaum wird jedoch davon ausgegangen, dass sich die Ausführungsinstanzen nicht unterscheiden. So ermöglichen die Replikationsforderungen, dass beispielsweise Ressourcen und Funktionen unabhängig von der Hardware immer bereit stehen (siehe [ST17, S. 9]). Da die Hardware der Ausführungsinstanzen bei den verteilten IoT-Anwendungen teilweise selbst Dienste zur Verfügung stellt, kann hier nicht von dieser Hardwareunabhängigkeit ausgegangen werden. Eine Funktion zur Steuerung von Hardware, ohne ihre Hardware ergibt logischerweise keinen Sinn. Auch der zu beachtende Datenschutz führt zu einer Bindung an bestimmte Geräte. Aus diesen Gründen unterscheiden sich die hier vorgestellten verteilten Anwendungen von denen nach Tanenbaum.

Die verteilten Anwendungen ähneln auch dem Konzept des (mobilen) Edge- bzw. Fog-Computing aus Unterabschnitt 2.3.2. Die Konzepte unterscheiden sich jedoch in wichtigen Punkten: Beim Edge-Computing wird ebenfalls davon ausgegangen, dass die Ausführungsinstanzen identisch und austauschbar sind. Anwendungen des Edge-Computing werden zudem in der Regel aus einer zentralen Sicht konzipiert (siehe Unterabschnitt 2.3.2). Bei verteilten Anwendungen für die hierarchische Kommunikationsstruktur stehen Datenschutz- und Datensicherheitsanforderungen der lokalen Anwenderinnen und Anwender im Vordergrund. Entsprechend werden Anwendungen hier nach einem Nutzer-zentrierten Ansatz konzipiert.

### 4.3.5 Einheitliche Softwareplattform

Damit die auf vielen, zumeist unterschiedlichen, Geräten verteilten Software-Elemente ausgeführt werden können, bedarf es gleicher Schnittstellen. Um auch eine mehrfache Nutzung der Software-Elemente selbst zu ermöglichen, sollen auch die Ausführungsumgebungen identisch sein. Für das System soll dazu eine einheitliche hardwareunabhängige Softwareplattform mit definierten Schnittstellen als gemeinsame Basis verwendet werden. Im Vergleich zu individuellen Lösungen führt eine Softwareplattform zu einer optimaleren Interoperabilität der Software-Elemente.

Eine solche einheitliche Softwareplattform muss folgende Aufgaben erfüllen:

- Software-Elemente müssen von der Softwareplattform ausgeführt, aber auch installiert, aktualisiert und durch die Nutzerin bzw. den Nutzer konfiguriert werden können. Für die Ausführung bedarf es einheitlicher universeller Schnittstellen. Anwendungsspezifische Merkmale sollten bei der Konzeption der Schnittstellen nicht beachtet werden.
- Des Weiteren dient die Plattform der transparenten Vermittlung von Daten innerhalb der hierarchischen Kommunikationsstruktur im Sinne einer verteilten Middleware (siehe Unterabschnitt 2.3.3). Die Plattform agiert dabei als Datenbroker, der Daten zwischen den Komponenten vermittelt.
- Auch der Verknüpfung zwischen den Daten und den Software-Elementen muss über die Softwareplattform erfolgen. Gleiches gilt auch für den Zugriff eines Software-Elements auf Funktionen eines anderen Software-Elements. Solche Verknüpfungen sollten auf Wunsch der Nutzerin bzw. des Nutzers und auf Anforderung eines Software-Elements erfolgen. Im Sinne des Auto Discovery (siehe Abschnitt 3.1) sollten solche Verknüpfungen auch automatisch erfolgen können.

Die Plattform soll als Basis auf allen beteiligten Komponenten, wie Servern, lokalen Koordinatoren und Geräten ausgeführt werden. Durch die hohe Wiederverwendbarkeit der Plattform kann auch davon ausgegangen werden, dass das Basissystem im Vergleich zu Einzellösungen selbst gehärtet und unanfälliger für Fehler und Sicherheitslücken wird. Insbesondere von dem Hintergrund zunehmender Angriffe auf IoT-Geräte ist dieser Aspekt nicht unbedeutend.

Bisher werden bei der Entwicklung von Anwendungsfällen im IoT für alle Plattformen (Geräte, Smartphones, Koordinatoren, Cloud-Dienste, usw.) unterschiedliche Schnittstellen und Programmiersprachen verwendet (siehe Unterabschnitt 2.3.2). Die Entwicklung unter

Heranziehung eines einheitlichen Anwendungskonzepts und sogar in nur einer Programmiersprache kann vermutlich den Implementierungsaufwand deutlich senken und gleichzeitig für eine höhere Skalierbarkeit der IoT-Anwendung sorgen.

## 4.4 Nutzerunterstützung für das Energiemanagement

Auf Basis der verteilten IoT-Anwendungen (siehe Unterabschnitt 4.3.4) und einer einheitlichen Softwareplattform können nun Energiemanagement-Anwendungen unter Einhaltung der Anforderungen an den Datenschutz und an die Datensicherheit unabhängig von der Art der Geräteeinbindung entworfen werden. Im Rahmen dieses Abschnitts werden Ansätze zur Nutzerunterstützung für die Realisierung von Energiesparmaßnahmen vorgestellt. Als Basis werden hierzu die zwei Funktionalitäten mit dem Fokus der Nutzerunterstützung aus Unterabschnitt 3.3.1 verwendet. Die Funktionalitäten adressieren verschiedene Zielgruppen, die sich jedoch teilweise überschneiden.

### 4.4.1 Nutzerunterstützung für Expertinnen und Experten sowie Eigentümerinnen und Eigentümer

Die Nutzergruppe der Eigentümerinnen und Eigentümer einer betreffenden Umgebung bzw. der Netzkunden verfolgt im Wesentlichen das Ziel, die Gesamtkosten zu kontrollieren bzw. zu senken. Hierzu bietet sich eine Top-Down-Betrachtung an. Von den Gesamtenergiekosten ausgehend werden zunächst die Geräte identifiziert, die die größten Kosteneinsparungen ermöglichen. Insbesondere für (externe) Expertinnen und Experten und zentrale Verantwortliche eines Unternehmens ist eine Übersicht der Verbräuche der Geräte sehr hilfreich, da ihnen die jeweiligen Umgebungen in der Regel zunächst nicht bekannt sind.

Die Abbildung 4.5 zeigt die grundsätzlichen Komponenten, die für eine Visualisierung von Verbrauchsinformationen benötigt werden. Hauptbestandteil ist hier ein Software-Element, das auf dem Smartphone, Tablet oder sonstigem Anzeigeeinterface (engl. User Interface, UI) der Zielgruppe ausgeführt wird. Dies wird in der Abbildung mit „App“ bezeichnet. Über die lokalen Koordinatoren (LK) erhält das Software-Element in regelmäßigen Abständen die anzuzeigenden Messdaten der Geräte (G). Die Messdaten können dabei auch von zusätzlichen Messgeräten stammen, auch wenn dies hier nicht explizit abgebildet ist. Auch die Nachbildung fehlender Informationen, wie in Unterabschnitt 4.1.4 beschrieben, kann hier berücksichtigt werden. Bei dieser Struktur wird davon ausgegangen, dass die Verbindung zwischen dem Anzeigegerät und den lokalen Koordinatoren nur dann zulässig und möglich ist, wenn sich das



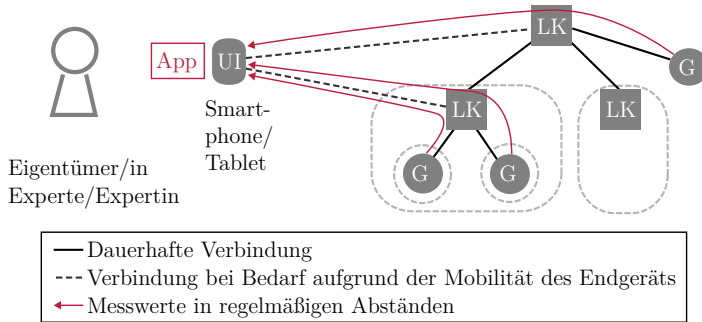


Abbildung 4.5: Struktur zur Visualisierung von Verbrauchsinformationen von Geräten. UI steht für Nutzerinterface (engl. User Interface), LK steht für die lokalen Koordinatoren und G für Geräte (vgl. Abschnitt 4.3). Die Messdaten werden in regelmäßigen Abständen an das Nutzerinterface gesendet. Initialisierungsbefehle sind nicht abgebildet, sie folgen aber den gleichen Pfaden wie die Messdaten.

Anzeigegerät in der lokalen Umgebung des lokalen Koordinators befindet. Daraus folgernd kann ein Smartphone oder Tablet durch seinen mobilen Einsatz mit unterschiedlichen lokalen Koordinatoren Verbindung aufnehmen. Das Software-Element des Anzeigegeräts kann dann jeweils die zur Verfügung stehenden relevanten Informationen abrufen und anzeigen. Die Regeln zur Wahrung des Datenschutzes und der Datensicherheit sorgen gleichzeitig dafür, dass Daten nur wenn erforderlich weiter kommuniziert werden.

Expertinnen und Experten sollten in die Lage versetzt werden, zusätzliche Analysen in der Nutzeroberfläche zu aktivieren. Diese Analysen müssen jedoch in der Struktur auf den lokalen Koordinatoren aktiviert und ausgeführt werden. Sie sollten ebenfalls als eigenständige Software-Elemente ausgeführt werden.

#### 4.4.2 Nutzerunterstützung für Anwenderinnen und Anwender

Die Anwenderinnen und Anwender bedienen die Geräte ihrer Umgebung. Sie kennen die Umgebung und ihre Geräteausstattung. Das Interesse der Anwenderinnen und Anwender ist, diese Geräte möglichst effizient zu nutzen.

Abbildung 4.6 zeigt die Struktur dieser Funktionalität. Dabei sind die grundsätzlichen Komponenten identisch mit Abbildung 4.5. Anstelle von Messdaten werden hier Ereignisse über die Gerätzustände an das Software-Element im Anzeigegerät übermittelt. Da nicht jedes Gerät Ereignisse liefert, kann hier eine Zustandsidentifikation wie in Abschnitt 4.2

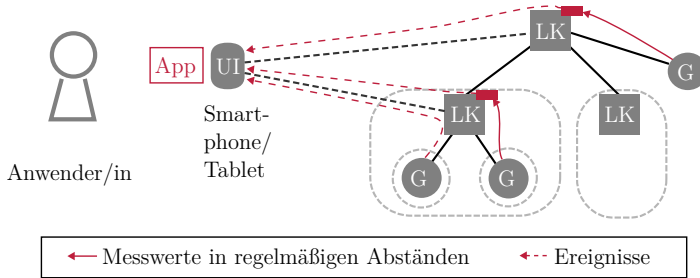


Abbildung 4.6: Struktur zur Darstellung von Informationen zu einzelnen Gerätenutzungen. Die roten Quadere symbolisieren eine Softwareverarbeitung.

verwendet werden. Diese Softwareverarbeitungen sind in der Abbildung als rote Quadere dargestellt.

#### 4.4.3 Navigationsstruktur für Nutzer-unterstützende Energiemanagement-Anwendungen

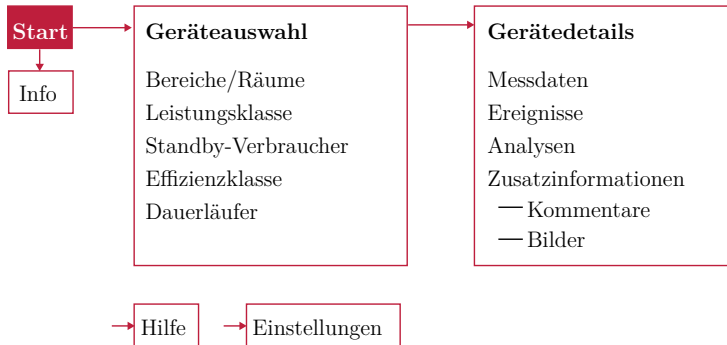


Abbildung 4.7: Navigationsstruktur für grafische Anzeigegeräte zur Anzeige von Energieverbrauchsdaten und -ereignissen

Das grafische Nutzerinterface für Expertinnen und Experten sowie Eigentümerinnen und Eigentümer (siehe Unterabschnitt 4.4.1) benötigt eine Navigationsstruktur, um die große Anzahl an Geräten, Messdaten und Zusatzinformationen übersichtlich präsentieren zu können. Damit Ereignisse nur nach Aufforderung an eine Anwenderin oder an einen Anwender weitergeleitet werden, sollte auch das grafische Nutzerinterface nach Abbildung 4.6 eine solche Navigationsstruktur zur Anzeige und Konfiguration aufweisen.

Abbildung 4.7 zeigt eine Navigationsstruktur, die auf eine Geräteauswahl und -details fokussiert. Die Abbildung zeigt verschiedene Vorschläge für die Geräteauswahl. Neben der Navigation über Räume oder Bereiche hinweg, könnte auch eine Navigation nach Leistungsklassen erfolgen. Auch eine Auswahl spezifischer Geräteklassen wäre möglich, wie z. B. Geräte mit Standby-Modus, Geräte mit schlechter Effizienz oder Geräte, die permanent im Betrieb sind (sogenannte Dauerläufer). Die Geräteauswahl könnte dabei auch mehrstufig erfolgen. Insbesondere bei einer Navigation nach Bereichen in großen Umgebungen ist dies sinnvoll.

Die Visualisierung der Gerätedetails sollte sowohl einen Überblick, aber auch die Auswahl von Detailinformationen ermöglichen. Für Expertinnen und Experten wäre die Verlaufsdarstellung verschiedener Messdaten aussagekräftig. Weiterhin sollte die Möglichkeit bestehen, zusätzliche Analysen aus den (Mess-)Daten zu aktivieren. Auch diese Analysen sollten visualisiert oder per Alarmmeldung an die Nutzerin bzw. den Nutzer weitergeleitet werden können (vgl. Kriterien in Unterabschnitt 2.2.4). Für Anwenderinnen und Anwender sollten zusätzlich die letzten Aktivitäten eines Gerätes visualisiert und mit Informationen, wie z. B. den verursachten Energiekosten, versehen werden.

## 4.5 Automatisiertes Energiemanagement

Neben den Nutzer-unterstützenden Funktionalitäten sollen, laut Unterabschnitt 3.3.1, auch Funktionalitäten zur automatisierten Steuerung von Energieverbrauchern und -erzeugern ermöglicht werden. In diesem Abschnitt werden Konzepte gezeigt, wie eine solche Optimierung in der hierarchischen Struktur (siehe Abschnitt 4.3) realisiert werden könnte.

### 4.5.1 Steuerung auf Basis einer direkten Verknüpfung

Wie Abbildung 4.8 verdeutlicht, wird bei dieser Funktionalität ein Gerät durch ein Software-Element (roter Quader) mit einem Steuerbefehl angesteuert, sobald ein zuvor verknüpftes Ereignis auftritt. In der Abbildung unter (a) handelt es sich dabei z. B. um den Status eines anderen Gerätes. Unter (b) wird dargestellt, dass sich eine solche Verknüpfung auch über mehrere lokale Koordinatoren hinweg erstellen lässt.

In Abbildung 4.8 unter (b) wird genauer ein Direkteingriff im Sinne des Market Demand Response (siehe Unterabschnitt 2.1.2) dargestellt. Hier fordert eine entfernte Partei direkt zu einer Abschaltung (oder auch zu einer Zuschaltung) auf. Dies findet in Form einer zweistufigen Kommunikation statt: Zuerst teilt das Gerät in einem Initialisierungsaufwurf mit,

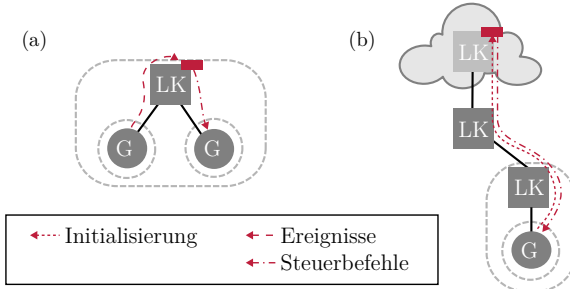


Abbildung 4.8: Struktur zur Steuerung auf Basis einer direkten Verknüpfung. Unter (a) wird eine direkte Geräteabstimmung gezeigt. Unter (b) wird eine Steuerung aufgrund eines Direkteingriffs im Sinne des Market Demand Response aus Unterabschnitt 2.1.2 realisiert.

wie viel elektrische Arbeit es zeitlich verschieben kann. Bei Bedarf übermittelt die entfernte Partei einen Steuerbefehl. Dabei wird davon ausgegangen, dass diese elektrische Arbeit nach der Wieder-Zuschaltung zusätzlich (bzw. bei Wieder-Abschaltung weniger) verbraucht wird. Damit beide Kommunikationsnachrichten durch die hierarchische Kommunikationsstruktur geleitet werden können, müssen auf jedem lokalen Koordinator die entsprechenden Zugriffsrechte eingeräumt werden.

Damit an dieser Stelle tatsächlich garantierte Abschaltungen sowie auch eine Entlohnung stattfinden können, müssen weitere technische und gesetzliche Rahmenbedingungen beachtet bzw. geschaffen werden. (siehe Unterabschnitt 2.1.1 und 2.1.2 zum Thema Präqualifikation)

#### 4.5.2 Steuerung auf Basis von Schwellenwerten



Abbildung 4.9: Struktur zur Steuerung auf Basis von Schwellenwerten. Diese kann für die Lastspitzenreduktion oder für die Steuerung auf Basis variabler Tarife verwendet werden.

Die Abbildung 4.9 fasst die Steuerung von Geräten auf Basis von Schwellenwerten zusammen.

Die Anwendung der Schwellenwerte wird dabei in einem Software-Element (rote Quader) vorgenommen. Die Wolke steht stellvertretend für eine beliebige Anzahl von lokalen Koordinatoren. Als Datenquellen kommen verschiedenste Daten in Frage. Im Unterschied zu Abbildung 4.8 sollte beachtet werden, dass die Software in der Datenquelle keine Informationen erhält und auch nicht entscheidet, welche Geräte geschaltet werden. Dies wird erst in der Struktur festgelegt.

Damit kann z. B. eine Lastspitzenreduktion (siehe Unterabschnitt 2.1.2) implementiert werden. Dazu wird zusätzlich eine Leistungsmessung in der Zuleitung zum Netzbetreiber benötigt. Mithilfe dieser Messdaten kann die voraussichtliche verbrauchte Energie der angebrochenen Viertelstunde (Betrachtungszeitraum des Leistungspreises) hochgerechnet werden. Der Quotient aus Vorhersage und gewünschtem Grenzwert dient der oben beschriebenen Struktur als Datenquelle. Die Schwellenwerte können individuell pro Gerät in Abhängigkeit von dessen Leistung und dessen Wichtigkeit eingestellt werden.

Eine Steuerung auf Basis variabler Tarife kann in ähnlicher Form erfolgen. Der aktuelle Energiepreis und ggfs. seine zukünftigen Veränderungen werden dabei als Datenquellen für die beschriebene Struktur verwendet. Anhand von individuellen Schwellenwerten können die Geräte dann einzeln angesteuert werden. Da der Energiepreis für alle benachbarten Geräte gleich ist, kann für jedes Gerät eine separate Entscheidung getroffen werden und eine Optimierung über mehrere Geräte hinweg ist nicht erforderlich.

### **4.5.3 Abstimmung von Geräten untereinander anhand von Optimierungsalgorithmen**

Energiemanagement-Algorithmen, wie sie in Unterabschnitt 2.2.2 vorgestellt wurden, sind eine weitere Möglichkeit, den Energieverbrauch einer Umgebung zu optimieren. Sie unterstützen Prosumer dabei, ihren Energieverbrauch auf ihre lokale Energieerzeugung anzupassen. Erzeugungsanlagen werden dabei genauso betrachtet wie Energieverbraucher, jedoch wird ihr Energiebeitrag mit umgekehrten Vorzeichen eingerechnet. Energiespeicher können je nach Energiefluss-Richtung negative, aber auch positive Vorzeichen aufweisen.

Die in Unterabschnitt 2.2.2 vorgestellten Algorithmen werden auf der lokalen zentralen Struktur des HEMS ausgeführt, in der alle relevanten Daten in einem System vorliegen (siehe Unterabschnitt 2.2.1). Bei verteilten Systemen ist dies nicht der Fall. Deshalb wird für verteilte Systeme die Struktur aus Abbildung 4.10 vorgeschlagen. Dabei tauschen Optimierungsverfahren über verschiedene Ebenen hinweg iterativ Prognosen aus. Die Daten werden folgendermaßen verarbeitet:

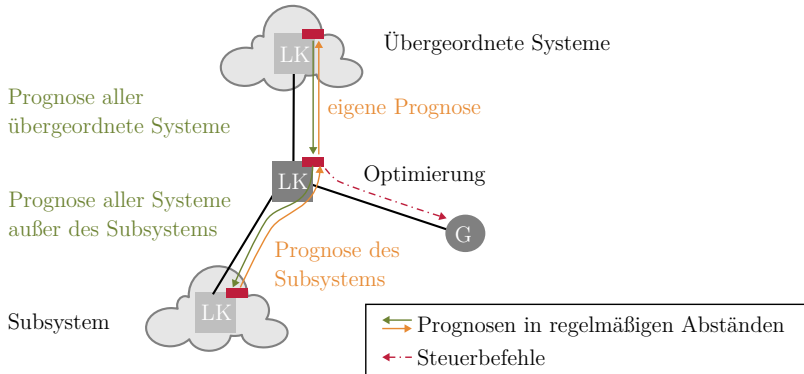


Abbildung 4.10: Struktur zur Abstimmung von Geräten (G) untereinander anhand von Optimierungsalgorithmen. Hier werden Prosumer dabei unterstützt, ihren Energieverbrauch und ihre lokale Energieerzeugung aufeinander abzustimmen. Die Wolken symbolisieren, dass sich hinter einem lokalen Koordinator noch weitere Koordinatoren und damit Ebenen befinden können.

- Die Summe aller erhaltenen fremden Prognosen wird für die Optimierung der lokalen Geräte verwendet. Daraus wird eine neue Prognose der lokalen Geräte berechnet.
- Anschließend wird für jedes Fremdsystem die Summe aller anderen Prognosen inklusive der neuen Prognose der Geräte gebildet und an dieses Fremdsystem versendet. In dieser Summierung wird die vom Fremdsystem erhaltene Prognose also nicht mit einbezogen.

Für die Optimierung sollten verschiedene Bewertungskriterien vorgegeben werden können. Dabei sollte auch eine Gewichtung mit variablen Energiepreisen möglich sein. Der beschriebene Prozessablauf findet in jedem System in jeder Hierarchieebene statt. Im obersten System wird naturgemäß keine Prognose von überliegenden Systemen eingerechnet.

## 4.6 Weitere Funktionalitäten jenseits des Energiemanagements

Nachfolgend wird dokumentiert, wie in den letzten Abschnitten anhand von Strukturabbildungen aufgezeigt, wie die in Unterabschnitt 3.3.1 vorgestellten Funktionalitäten in der verteilten Struktur abgebildet werden können. In diesem Abschnitt werden dabei Funktionalitäten jenseits des Themenfeldes des Energiemanagements fokussiert.



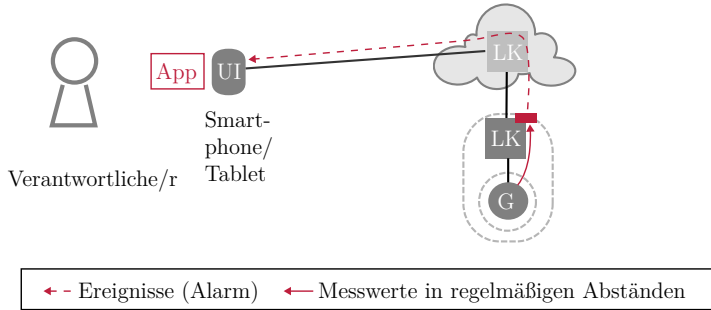


Abbildung 4.12: Struktur für die prädiktive Instandhaltung anhand einer Überwachung der Messdaten von Geräten. Bei einem Fehler wird eine Alarmmeldung an die Verantwortliche bzw. den Verantwortlichen gesendet.

### 4.6.3 Sicherheitsabschaltungen

Im Folgenden sollen zwei verschiedene Sicherheitsabschaltungen beschrieben werden. Beide gehen ähnlich zu Unterabschnitt 4.5.1 davon aus, dass aus einem Ereignis ein Steuerbefehl abgeleitet wird.

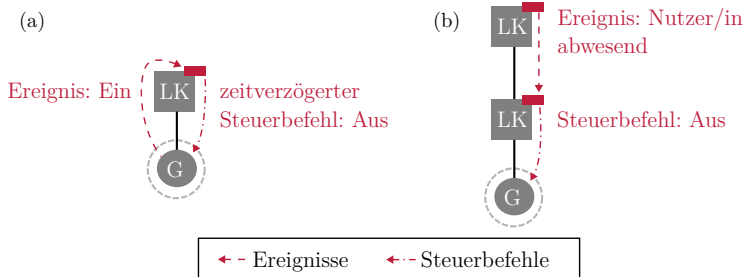


Abbildung 4.13: Sicherheitsabschaltungen in Abhängigkeit einer Zeitdauer oder der Anwesenheit von Bewohnerinnen und Bewohnern

Im Fall (a) von Abbildung 4.13 wird ein Gerät nach einer festen Zeitspanne automatisch abgeschaltet. Im Fall (b) wird der Steuerbefehl aus einem sonstigen Ereignis abgeleitet, hier aus der erkannten Abwesenheit der Bewohnerinnen und Bewohner.



## 5 Realisierung eines Systems zum Energiemanagement

Damit die Praxisfähigkeit der in Kapitel 4 dargelegten Konzepte in realen Umgebungen untersucht und demonstriert werden kann, wurde das Systemkonzept für Energiemanagementsysteme in Form verschiedener Software-Komponenten realisiert. Bei allen Implementierungen wurde darauf geachtet, diese möglichst auf ihre Grundbestandteile und -konzepte zu reduzieren. Dadurch sind universell erweiterbare Lösungen entstanden.

Wie in Unterabschnitt 4.3.5 vorgeschlagen, wurde zunächst eine Softwareplattform realisiert, die als einheitliche Basis allen beteiligten Geräten zur Ausführung von (verteilten) Anwendungen dient. Diese Plattform wird nachfolgend auch als *IoT-Softwareplattform* bezeichnet. Sie soll die Ausführung von verteilten Anwendungen auch in anderen Anwendungsbereichen erlauben. Die Aufgaben und Verfahren der oberen und unteren Schicht des Systemkonzepts, also der Geräteeinbindung (siehe Abschnitt 4.1 und 4.2) und der Funktionalitäten des Energiemanagements (siehe Abschnitt 4.4 bis 4.6), werden in der Realisierung nicht getrennt voneinander, sondern gleichwertig als Anwendungen der IoT-Softwareplattform umgesetzt. Auf diese Plattform und die Realisierung der Kommunikation wird in Abschnitt 5.1 eingegangen. Insbesondere bei der Kommunikation wurde darauf geachtet, keine anwendungsspezifischen Strukturen zu implementieren. So können durch die allgemeinen Schnittstellen auch viele andere Anwendungen auf Basis der IoT-Softwareplattform implementiert werden.

Die größte Herausforderung für die Etablierung eines feingranularen Energiemanagements stellt die Einbindung einer möglichst großen Anzahl an Geräten dar. Auf die Realisierung von Geräteeinbindungen mit Hilfe der IoT-Softwareplattform wird ebenfalls in Abschnitt 5.1 eingegangen. Wie bereits in Abschnitt 4.1 und 4.2 erörtert, unterscheiden sich Geräte in vielen Dimensionen voneinander, weshalb unterschiedliche Erkennungs- und Nachbildungsverfahren erforderlich sind. In der Praxis ist nur ein Teil der heutigen Geräte bereits mit geeigneten Datenschnittstellen versehen. Deshalb wurden in Unterabschnitt 4.1.3 bzw. 4.2.3 und folgende verschiedene Verfahren vorgestellt, um fehlende Informationen nachzubilden. Basierend auf diesen Verfahren werden in Abschnitt 5.2 konkrete Umsetzungen vorgestellt, um die Geräteeinbindungen zu ermöglichen oder mit weiteren Informationen anreichern.

In Abschnitt 5.3 werden exemplarische Anwendungen vorgestellt, die hauptsächlich dem Energiemanagement dienen. Sie wurden aus den Konzeptionen aus Abschnitt 4.4 bis 4.6 abgeleitet. Unter den Anwendungen finden sich Anwendungen zur Nutzerunterstützung und automatisierte Energiemanagementanwendungen. Darüber hinaus werden auch Funktionalitäten jenseits des Energiemanagements umgesetzt.

Die in diesem Kapitel benannten Anwendungen (hervorgehoben durch Unterstreichungen) sind unter <https://osiota.net/> als Open-Source-Software verfügbar. Der Name osiota steht dabei für *Operating System for Internet of Things Applications* (engl. für System zur Ausführung von Anwendungen des Internet der Dinge) und bezeichnet die Basiskomponente in Form der IoT-Softwareplattform. Weitere Anwendungen sind in getrennten Repositories abgelegt, die nach ihren Anwendungsnamen benannt sind.

## 5.1 Umsetzung der IoT-Softwareplattform

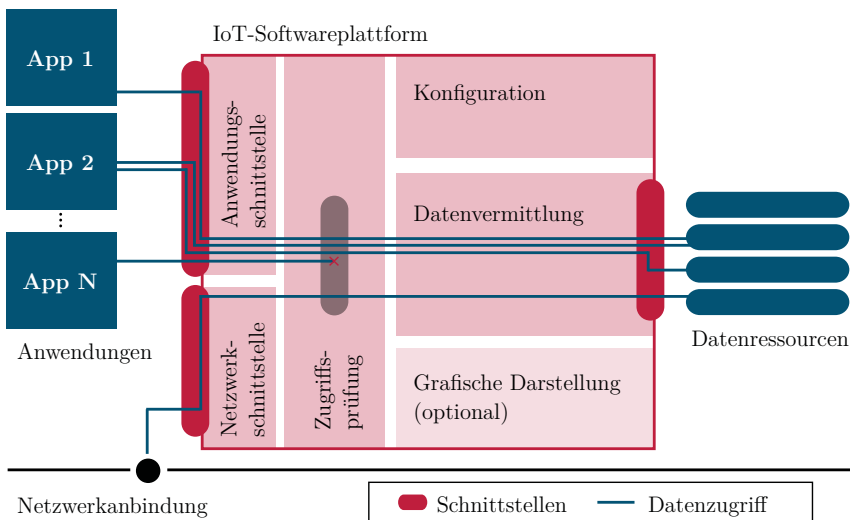


Abbildung 5.1: Struktur der Implementierung unterteilt in Anwendungen, Netzwerkanbindung und IoT-Softwareplattform. Damit die Anwendungen Daten untereinander austauschen können, stellt die IoT-Softwareplattform den Anwendungen sogenannte Datenressourcen bereit. Auch ein Austausch über das Netzwerk ist mit Hilfe der Datenressourcen möglich. Folglich finden sich universelle Schnittstellen zu den Anwendungen, zu den Datenressourcen und für die Netzwerkanbindung.

In Unterabschnitt 4.3.5 wurden die Aufgaben einer Softwareplattform für verteilte Anwendungen skizziert. Hierzu zählen die Ausführung von Anwendungen, der Austausch von Daten sowie die Verknüpfung mit Daten und Funktionen anderer Anwendungen. Für diese Aufgaben werden universelle Schnittstellen benötigt. In Abbildung 5.1 wird hierzu die Struktur der Implementierung und der notwendigen Schnittstellen dargestellt. Abgebildet sind drei Anwendungen, die IoT-Softwareplattform und eine Netzwerkanbindung, über welche die IoT-Softwareplattform Daten mit anderen Instanzen der IoT-Softwareplattform austauschen kann. Ein Modul innerhalb der IoT-Softwareplattform sorgt für den Zugriffsschutz. Daten werden als sogenannte *Datenressourcen* gehalten. Diese Datenressourcen implementieren die in Unterabschnitt 4.3.1 beschriebenen Datenobjekte. Für die eindeutige Adressierbarkeit beinhalten die Datenressourcen im Gegensatz zu den Datenobjekten jedoch immer nur einen Datenwert. Demzufolge werden mehrere Datenressourcen benötigt, um beispielsweise die Daten eines Gerätes abzubilden.

Aus der Abbildung 5.1 werden die Schnittstellen deutlich, die sich zwischen den Elementen befinden:

- Die Schnittstelle zu den Datenressourcen ist zentral in der Mitte dargestellt. Zur Interaktion mit den Datenressourcen wurden **Basismethoden** basierend auf dem Publish-Subscribe-Entwurfsmuster (siehe Unterabschnitt 2.3.3, vgl. [Eug+03]) ausgearbeitet. Diese Schnittstelle bedient die zuvor benannte Aufgabe des Datenaustauschs. Zusätzlich wird durch die Basismethoden auch die Aufgabe der Verknüpfung von Anwendungen mit Daten und Funktionen anderer Anwendungen umgesetzt.
- Erzeugt und genutzt werden die Datenressourcen durch die Anwendungen, die mittels einer **Anwendungsschnittstelle** auf die Datenressourcen zugreifen. Über diese Anwendungs-Schnittstelle werden die Anwendungen auch gestartet, konfiguriert und wenn notwendig wieder beendet.
- Zum Aufbau der hierarchischen Kommunikationsstruktur ist eine Netzwerkanbindung mit einer eigenen **Netzwerkschnittstelle** vorgesehen. Diese Schnittstelle stellt u. a. die Basismethoden zur Interaktion mit den Datenressourcen bereit. Wenn erforderlich, können damit Datenressourcen mit anderen Instanzen der IoT-Softwareplattform ausgetauscht werden. Zusätzlich werden weitere Methoden für die Initialisierung und Überprüfung der Netzwerkverbindung bereitgestellt.

In der Abbildung ist lediglich jeweils eine Kommunikationsschnittstelle zwischen den Anwendungen bzw. dem Netzwerk und den Datenressourcen angedeutet. Eine direkte Kommunikation zwischen der Netzwerkanbindung und den Anwendungen ist nicht vorgesehen. Auch

Anwendungen können nicht direkt miteinander kommunizieren. Jede Art der Kommunikation zwischen den Elementen muss ausschließlich über die Datenressourcen erfolgen.

Das Grundsystem und die meisten Anwendungen wurden in der Skriptsprache JavaScript erstellt und werden mit dem Interpreter Node.js ausgeführt. Mit Node.js können hochperformante Netzwerkanwendungen ermöglicht werden (s. [TV10]). Die Abhängigkeiten zu externen Quelltextbibliotheken werden mit der Node.js-eigenen Paketverwaltung NPM (für Node.js Packet Manager, engl.) organisiert.

Jede Instanz wird durch mindestens eine Instanz-Konfigurationsdatei beschrieben. Beim Start der Instanz muss die Konfigurationsdatei angegeben werden. Die Datei beinhaltet die Definition, welche Anwendungen gestartet werden sollen sowie die zugehörigen Konfigurationsdaten. Auch die IoT-Softwareplattform selbst kann über diese Konfigurationsdatei konfiguriert werden.

Die Verwendung der Datenressourcen wird nachfolgend in Unterabschnitt 5.1.1 erläutert. Die dabei verwendeten Basismethoden werden anschließend in Unterabschnitt 5.1.2 beschrieben. Die Anwendungs-Schnittstelle zwischen der IoT-Softwareplattform und den Anwendungen wird in Unterabschnitt 5.1.3 erläutert. Die Beschreibung der Netzwerk-Schnittstelle und der verwendeten Kommunikationstechniken sind Unterabschnitt 5.1.4 zu entnehmen. Um konkrete Anforderungen an den Datenschutz und die Datensicherheit erfüllen zu können, wird in Unterabschnitt 5.1.5 ein Mechanismus zur Begrenzung der Weiterleitung von Datenressourcen beschrieben.

In Unterabschnitt 5.1.6 wird eine Erweiterung zur Darstellung von Nutzerschnittstellen mittels HTML5 vorgestellt. Dadurch können alle Bestandteile einer Funktionalität auf Basis der IoT-Softwareplattform realisiert werden. Eine Auswahl an implementierten Anwendungen für die IoT-Softwareplattform wird abschließend in Unterabschnitt 5.1.7 vorgestellt.

Die Realisierung der Geräteeinbindung auf Basis der IoT-Softwareplattform wird in Unterabschnitt 5.1.8 erläutert. Dabei wird auf die zuvor erläuterten Komponenten eingegangen. Zur Nachbildung des Verhaltens eines nicht-vernetzten Gerätes ist eine Geräterepräsentation erforderlich (siehe Abschnitt 4.2). Hierzu wird eine universelle Software-Anwendung für die IoT-Softwareplattform vorgestellt. Mittels Vererbung können andere Software-Anwendungen Funktionen dieser universellen Software-Anwendung verwenden.

### 5.1.1 Datenressourcen: Austausch und Speicherung von Informationen

*Datenressourcen* dienen innerhalb der IoT-Softwareplattform zur Bereitstellung von Informationen. So können Daten zwischen den Anwendungen bzw. zwischen den Anwendungsteilen

bereitgestellt und gegenseitige Reaktionen durchgeführt werden. Als Datentypen können Zahlen, Zeichenketten, bool'sche Werte sowie Objekte und Arrays aus diesen Datentypen (also alle JavaScript-Datentypen, jedoch keine Funktionsreferenzen und Objektmethoden) verarbeitet werden. Der letzte Wert wird stets im Feld **value** abgelegt. Zusätzlich wird auch der Zeitstempel der letzten Änderung der Informationen im Feld **time** in den Datenressourcen abgelegt.

Weiterhin besitzen die Datenressourcen jeweils einen lokal (für die Instanz) eindeutigen Namen, der nicht veränderbar im Feld **name** abgelegt ist. Über diesen Namen kann eine Ressource adressiert werden. Die Ressourcen können dabei hierarchisch organisiert werden. Der Name besteht dann aus einer Pfadangabe, die zusätzlich um den Namen der übergeordneten Datenressourcen ergänzt wird. Als Trennzeichen kommen Schrägstriche („/“) zum Einsatz. Als Beispiel kann eine Datenressource den Namen „Wohnung / Küche / Geschirrspülmaschine / Energieverbrauch“ tragen. Eine Anwenderin oder ein Anwender kann aus dem Namen schließen, dass die Datenressource Angaben zum Energieverbrauch des Gerätes Geschirrspülmaschine beinhaltet und den Ort des Gerätes ermitteln. Für die Software besitzen die Bezeichnungen jedoch keine sinngebende Bedeutung.

Für eine sinngebende Bedeutung werden die Datenressourcen deshalb mittels zusätzlicher Metadaten beschrieben, wie dies bereits in Unterabschnitt 4.3.1 vorgeschlagen wurde.

- Die Metadaten dienen insbesondere zur **Beschreibung** der Daten, die innerhalb der Datenressourcen abgespeichert sind. Dadurch sind Anwendungen in der Lage, die Daten entsprechend zu verarbeiten bzw. anzuzeigen. Im Fall eines Zahlenwerts könnte dies beispielsweise die Ergänzung um die physikalische Einheit und den zulässigen Wertebereich sein. So könnte beispielsweise beim Energiemanagement eine Anwendung auf die Bekanntgabe der Leistungsaufnahme in den Metadaten einer Datenressource eines repräsentierten Gerätes reagieren.
- Weiterhin dienen die Metadaten zur **Identifizierung** der Datenressourcen. So könnte in den Metadaten der Ort in Form von Koordinaten, der Gerätetyp oder auch der Anwendungsname der erzeugenden Anwendung abgelegt sein. Dadurch sind Anwendungen in der Lage, bei Bedarf für sie interessante Datenressourcen selbstständig zu identifizieren. Diese Fähigkeit adressiert die Aufgabe der Verknüpfung von Anwendungen mit Daten und Funktionen anderer Anwendungen, wie sie in der Einleitung zu Abschnitt 5.1 erwähnt wird.

Die Metadaten sind als Schlüssel-Wert-Paare im Feld **metadata** organisiert. So können unter beliebigen Schlüsseln Daten oder ganze Strukturen abgelegt werden. Damit kann beispielsweise die Existenz oder die Wertübereinstimmung bestimmter Schlüssel überprüft

werden. Im Projekt Hypercat wurde ein Vorschlag für solche Schlüssel-Wert-Paare für das Internet der Dinge ausgearbeitet (vgl. [BL14]), der hier angewendet werden kann. In der Literatur finden sich weiterhin auch allgemeine Beschreibungen von realen Objekten in der Form von Datenstrukturen (z. B. [DB16], [Spo+14]), die grundsätzlich für die Metadaten adaptiert werden können. Da die Untersuchung der Metadaten nicht im Fokus dieser Arbeit steht, verwenden die später beschriebenen Anwendungen lediglich einen begrenzten Umfang an teils anwendungsspezifischen Schlüsseln und Werten.

### 5.1.2 Basismethoden zur Interaktion mit Datenressourcen

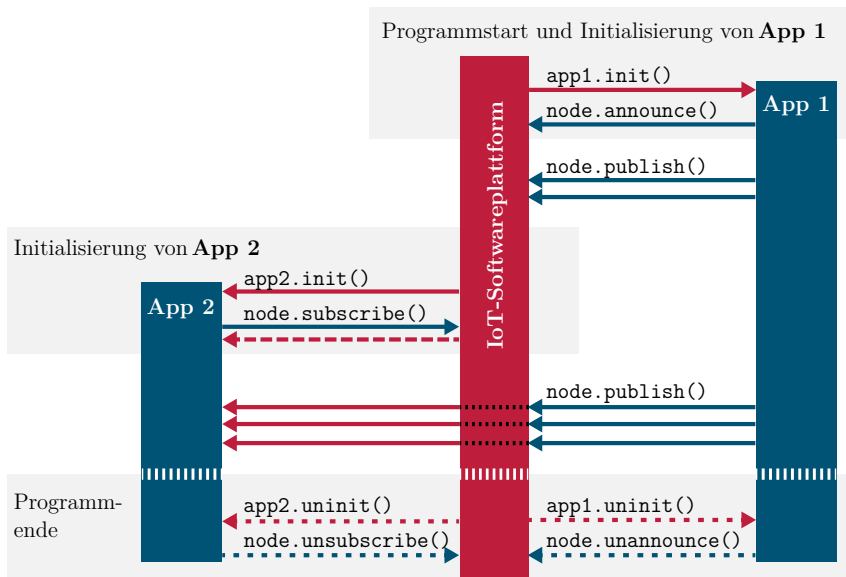


Abbildung 5.2: Initialisieren zweier Anwendungen durch die IoT-Softwareplattform. App 1 zeigt die Veröffentlichung einer Datenressource `node`. App 2 zeigt das Abonnieren einer fremden Datenressource `node`. Hierbei soll es sich um die gleiche Datenressource handeln. In Blau werden die Basismethoden dargestellt, die von den Anwendungen auf den Datenressourcen ausgeführt werden. In Rot sind die Initialisierungsaufrufe der IoT-Softwareplattform sowie die Weiterleitung der Datenveröffentlichungen durch die IoT-Softwareplattform (mit Hilfe von Callbacks) abgebildet.

Die in diesem Unterabschnitt vorgestellten Methoden werden im Rahmen dieser Arbeit als Basismethoden bezeichnet. Sie interagieren mit den zuvor beschriebenen Datenressourcen (siehe Unterabschnitt 5.1.1).

Die folgenden zwei Methoden setzen die Aufgabe des Austauschs von Daten um (siehe Abschnitt 5.1). Wie in Unterabschnitt 4.3.1 vorgeschlagen, wird dazu das Publish-Subscribe-Entwurfsmuster verwendet. Der Aufruf der beschriebenen Methoden ist exemplarisch in Abbildung 5.2 dargestellt.

- Eine Anwendung (hier App 2) kann bei Bedarf die Daten und deren zukünftige Aktualisierungen einer Datenressource mit der Methode `node.subscribe(callback)` abonnieren. Dabei muss eine Funktion `callback` übergeben werden, die bei dem Empfang neuer veröffentlichter Daten aufgerufen wird. Eine solche Funktion wird englischsprachig als `Callback` bezeichnet.
- Der Eigentümer einer Datenressource (hier App 1) kann mittels der Methode `node.publish(time, value)` neue Daten `value` in der Datenressource veröffentlichen. Dabei kann optional ein Zeitstempel `time` übergeben werden; anderenfalls wird ein neuer Zeitstempel erzeugt. Die Abonnenten dieser Datenressource werden durch den Aufruf der `Callback`-Funktionen über jede einzelne Veröffentlichung informiert.

Die ersten zwei Aufrufe der Methode `publish` in Abbildung 5.2 werden zunächst nicht weitergeleitet, da bislang kein Abonnement vorhanden ist. Erst nach der Initialisierung eines Abonnements durch App 2 werden fortan die Daten weitergeleitet. Die Aufrufe der `Callbacks` sind in der Abbildung ohne Funktionsnamen dargestellt. Direkt nach der Initialisierung des Abonnements werden zusätzlich auch die Daten der letzten Veröffentlichung mit Hilfe des `Callbacks` weitergeleitet. Dieser Aufruf ist in der Abbildung mit dem gestrichelten Pfeil direkt unter dem Aufruf der Methode `subscribe` angedeutet.

Zur Identifikation von Datenressourcen sollen neue Datenressourcen und ihre Metadaten bekannt gemacht werden. Auch hier wird auf das Publish-Subscribe-Entwurfsmuster zurückgegriffen, damit Anwendungen diese Bekanntmachungen abonnieren können. Mit den folgenden Methoden wird diese Fähigkeit umgesetzt:

- Mit der Methode `node.subscribe_announcement(callback)` können Bekanntmachungen von neuen Datenressourcen unterhalb der Datenressource `node` abonniert werden. Auch hierzu muss eine Funktion `callback` übergeben werden, die aufgerufen wird, wenn neue Datenressourcen bekannt gemacht werden. Diese Methode ist nicht in der Abbildung dargestellt. Sie stellt aber eine Möglichkeit dar, wie App 2 an die Instanz der Datenressource `node` gelangen kann.
- Weiterhin können mit der Methode `node.announce(metadata)` Datenressourcen von einer Anwendung, hier App 1, bekannt gemacht werden. Bei der Ankündigung werden

die Metadaten `metadata` (siehe Unterabschnitt 5.1.2) versendet, die die Datenressourcen beschreiben. Auch Änderungen der Metadaten können auf diesem Weg bekannt gegeben werden.

Zusätzlich finden sich folgende Methoden:

- Mittels der Methoden `node.unsubscribe()` bzw. `node.unannounce()` können das Abonnement bzw. die Veröffentlichung der Datenressource aufgehoben werden. In Analogie dazu kann mit der Methode `node.unsubscribe_announcement()` auch das Abonnement der Ankündigungen aufgehoben werden. In der Abbildung 5.2 werden diese Methoden beim Programmende verwendet. Diese sind gestrichelt dargestellt, da sich die Plattform als Dienst durchgängig in Funktion befindet und die Methoden bei diesem Einsatzzweck nur selten aufgerufen werden.
- Über die Methode `node.get_history(config)` können vorherige Daten als Zeitreihen abgefragt werden. Über das Konfigurationsobjekt `config` kann dabei die Anzahl der historischen Daten, das Startdatum, das Enddatum sowie die Datenauflösung eingeschränkt werden. Die IoT-Softwareplattform bietet verschiedene Möglichkeiten an, die historischen Daten abzuspeichern, u. a. im Arbeitsspeicher und in Dateien.

### 5.1.3 Schnittstelle für Software-Anwendungen

Die IoT-Softwareplattform kann zur Erweiterung ihres Funktionsumfangs Anwendungen laden. Bei den Anwendungen handelt es sich um abgetrennte, eigenständige Programmteile. Eine Anwendung kann aus einem oder auch aus mehreren Software-Elementen bestehen, wie sie in Unterabschnitt 4.3.4 beschrieben wurden. Jede Anwendung muss einzeln initialisiert werden. Die bereits benannte Anwendungs-Schnittstelle beinhaltet dazu folgende Methoden:

- Soll eine Anwendung gestartet werden, wird durch die IoT-Softwareplattform die Methode `app.init(node, app_config)` der Anwendung aufgerufen. Sofern Konfigurationsdaten vorliegen, werden diese mit der Methode als `app_config` mitübergeben. Zusätzlich wird auch eine individuelle Datenressource `node` übergeben, die von der Anwendung bei Bedarf genutzt werden kann. Wie in Unterabschnitt 5.1.2 beschrieben, muss die Datenressource dabei zunächst mit einem Aufruf der Methode `node.announce` initialisiert werden.
- Weitere Methoden, wie `app.uninit()` oder `app.reinit()` ermöglichen das Beenden bzw. das Neuladen von Anwendungen. Beim Neuladen kann auch eine neue veränderte Konfiguration übergeben werden.



- Über die Schnittstelle können die Anwendungen auch weitere Datenressourcen adressieren und diese mit den in Unterabschnitt 5.1.2 beschriebenen Methoden erzeugen bzw. abonnieren.
- Weiterhin kann eine Anwendung einer eigenen Datenressource mit der Methode `node.on_rpc(method, function)` weitere sogenannte RPC-Funktionen hinzufügen, die der entfernten Ausführung dienen (eng. remote procedure calls, RPC). Der jeweiligen RPC-Funktion `function` wird dabei ein Funktionsname `method` zugeordnet. Die RPC-Funktionen können von anderen Anwendungen unter Nennung dieses Funktionsnamens `method` mit Hilfe der Methode `node.rpc(method, arguments, [callback])` der Datenressource `node` aufgerufen werden. Dabei können weitere Argumente übergeben werden. Die RPC-Funktion wird mit diesen Argumenten im Kontext der ursprünglichen Anwendung ausgeführt. Die bei der Ausführung der RPC-Funktion zurückgegebenen Daten werden der optionalen Funktion `callback` übergeben. Damit stehen die Daten wieder der aufrufenden Anwendung zur Verfügung.

Für eine erfolgreiche Initialisierung einer Anwendung muss mindestens die Methode `init()` definiert sein. Daneben kann eine Anwendung auch beliebige andere Methoden definieren.

Anwendungen können Funktionen von anderen Anwendungen erben und sie als eigene Funktionen verwenden. Auch eine mehrfache Vererbung ist möglich. Dazu müssen ein oder mehrere Anwendungsnamen im Attribut `inherit` der Anwendung angegeben werden. Einzelne geerbte Funktionen können bei Bedarf überschrieben werden.

Die Anwendungen können über die Node.js-eigene Paketverwaltung NPM (siehe Abschnitt 5.1) automatisch installiert und auch aktualisiert werden. Dazu sollten die Anwendungen jeweils in eigenen GIT-Repositories verwaltet werden.

Das Konzept der Schnittstelle wurde gemeinsam mit einem Studenten in der Masterarbeit [16/014] entwickelt.

### 5.1.4 Netzwerkkommunikation

Wie gefordert, wird durch die Bereitstellung einer geeigneten Netzwerkkommunikation die Möglichkeit geschaffen, dass verschiedene Instanzen der IoT-Softwareplattform und deren Anwendungen auch über ein Netzwerk miteinander interagieren können. Um die Basismethoden abbilden zu können, muss das zugrunde-liegende Protokoll sowohl die Aufrufe des Publish-Subscribe-Entwurfsmusters als auch Aufrufe von RPC-Methoden transportieren können. Die in Unterabschnitt 2.3.4 beschriebenen Protokolle unterstützen jeweils nur einen dieser Anwendungsfälle. Die Kommunikationsprotokolle REST und CoAP beherrschen das

Publish-Subscribe-Entwurfsmuster nicht. Hingegen beherrscht das Kommunikationsprotokoll MQTT zwar das Publish-Subscribe-Entwurfsmuster, jedoch kann das Ausführen entfernter Funktionen nicht über Umwege realisiert werden<sup>1</sup>. Auch Funktionen zum Auto-Discovery sind in keinem der in Unterabschnitt 2.3.4 beschriebenen Protokolle vorgesehen<sup>2</sup>.

Um möglichst flexibel bei der Erprobung der Funktionsmechanismen der hierarchischen Kommunikationsstruktur sein zu können, wurde keines der vorhandenen Protokolle adaptiert, sondern ein neues Protokoll auf Basis der in Unterabschnitt 5.1.2 beschriebenen Basismethoden abgeleitet. Hierbei wurde auf die HTTP-Erweiterung WebSockets als Übertragungskanal zurückgegriffen. WebSockets bieten eine bidirektionale TCP/IP-basierte Paket-orientierte Kommunikation (siehe [FM11]). Über diese Verbindung werden JSON-kodierte (siehe [Bra17]) Pakete versendet, die jeweils eine entfernte Methode adressieren. Die Pakete enthalten den aufzurufenden Methodennamen und die Argumente der Methode. Neben den Basismethoden stehen zusätzlich folgende Methoden zur Verfügung:

- Über die Methode `hello(client_name, credentials)` wird die Verbindung vom Client aus initialisiert. Hier kann die Client-Instanz ihren Namen und Informationen zur Authentifizierung übertragen. Der Server antwortet daraufhin mit seinem Namen.
- Die Funktion `ping()` ermöglicht einen Verbindungstest. Der Server antwortet mit dem Text „ping“.
- Über die Hilfsfunktion `reply()` werden Rückgabewerte von Methoden zurück zur Gegenstelle übermittelt.

Beim Aufruf der Basismethoden muss zusätzlich die Datenressource benannt werden, auf welcher die Methode ausgeführt werden soll. Mit Ausnahme der Methode `hello` stehen alle Methoden sowohl in der Client- und auch in der Server-Instanz zum entfernten Aufruf bereit.

Das Abonnement von entfernten Datenressourcen wird in der IoT-Softwareplattform transparent und automatisch durchgeführt, sodass die Anwendungen nicht zwischen lokalen und entfernten Datenressourcen unterscheiden müssen. Dazu legt die IoT-Softwareplattform automatisch inhaltslose Kopien der entfernten Datenressourcen an, wenn diese über die Methode `subscribe_announcement` der lokalen Instanz bekannt gegeben werden. Anfänglich werden lediglich die Metadaten in diesen Kopien abgelegt. Erst wenn mindestens eine

---

<sup>1</sup>Zwar kann eine Funktion an eine MQTT-Ressource gebunden werden. Ob ein Befehl korrekt übermittelt wurde und mit welchem Ergebnis die Funktion ausgeführt wurde, kann dabei nicht ermittelt werden. Dazu kann eine zweite MQTT-Ressource verwendet werden, jedoch gibt es für die Benennung dieser Ressource keine einheitliche Konvention.

<sup>2</sup>Das Kommunikationsprotokoll SOAP bietet mit der Ergänzung WSDL (siehe Unterabschnitt 2.3.4) die Möglichkeit, Schnittstellen zu beschreiben. Dies könnte ggfs. für eine Auto-Discovery-Funktion adaptiert werden.

Anwendung eine Datenressource abonniert oder eine andere entfernte Instanz nach den Daten verlangt, werden auch die Daten dieser Datenressource über die Netzwerkverbindung abonniert. Auch die Aufrufe der RPC-Funktionen einer entfernten Datenressource werden automatisch über das Netzwerk weitergeleitet und die Rückgaben zurückübermittelt. Natürlich ist auch ein mehrfaches Weiterleiten von Datenressourcen über mehrere Instanzen der IoT-Softwareplattform möglich.

Der Server-Dienst kann durch die Anwendung **ws-server** konfiguriert und gestartet werden. Dabei muss eine Port-Nummer für den Server-Dienst angegeben werden. Mit Hilfe der Anwendung **wss-secure** kann auch ein Serverdienst mit SSL-Verschlüsselung auf Basis von HTTPS gestartet werden. Durch die Anwendung **ws** können auch mehrere Clientverbindungen initialisiert werden. Dabei muss eine Server-Adresse angegeben werden, mit der sich der Client verbinden soll. Diese setzt sich aus dem Text „ws://“ für unverschlüsselte bzw. „wss://“ für verschlüsselte Verbindungen, dem Gerätenamen bzw. der IP-Adresse sowie der Portnummer abgetrennt durch einen Doppelpunkt zusammen. Dies ergibt beispielsweise die Adresse „ws://192.168.1.2:8080“ für den unverschlüsselten Dienst auf dem Host 192.168.1.2 mit dem Port 8080.

### 5.1.5 Zugriffsschutz: Begrenzung der Weiterleitung von Datenressourcen

Wie in Unterabschnitt 4.3.3 erläutert, sollen nicht jede Datenressource und ihre Funktionen an entfernte Instanzen weitergeleitet werden. Um konkrete Anforderungen an den Datenschutz und die Datensicherheit einhalten zu können, prüft die IoT-Softwareplattform vor jeder Interaktion die Gestattung des Zugriffs auf die Datenressourcen. Ebenfalls wird auch der Zugriff durch Anwendungen überprüft und begrenzt. Wie in Unterabschnitt 4.3.3 vorgeschlagen, wird die Überprüfung anhand von zuvor definierten Regelsätzen durchgeführt.

Die Regelsätze können anhand des Namens, des Namensbereichs, der Metadaten, der Datenquelle (Anwendung oder entfernte Instanz) und bzw. oder der anfragenden Seite (Anwendung oder entfernte authentifizierte Instanz) verschiedene Aktionen definieren. Abweichend von den Vorschlägen für Regelsätze aus Unterabschnitt 4.3.3 sind hier die Vorverarbeitung und Verschlüsselung nicht mehr Teil der Regelsätze. Sie können aber beispielsweise durch die folgenden zwei Anwendungen realisiert werden. Die Anwendung **aggregation** erzeugt eine Datenressource mit Daten in aggregierter Form, in der mehrere Datenressourcen gleichen Typs zusammengefasst werden. Dabei ist auch eine zeitliche Akkumulation durch Resampling möglich. Die Anwendung **encryption-pubkey** verschlüsselt bzw. entschlüsselt Inhalte durch eine Public-Key-Verschlüsselung.

Die nachfolgenden Aktionen können bei der Weiterleitung berücksichtigt werden:

- a) **hide\_all – Keine Weiterleitung:** Die Ressource bleibt für die anfragende Seite verborgen.
- b) **hide\_value / hide\_value\_and\_metadata – Weiterleitung von Namen und Metadaten:** Die Existenz der Datenressource wird der anfragenden Seite mitgeteilt. Hierbei sind eine Datenweiterleitung und ein Aufruf von Funktionen nicht vorgesehen. Die Metadaten können bei **hide\_value\_and\_metadata** optional auf eine Auswahl eingeschränkt werden.
- c) **forward\_all – Vollständige Weiterleitung von Namen, Metadaten und Daten:** Wie b), jedoch zusätzlich mit Weiterleitung der Daten.

Der Aufruf der RPC-Funktionen kann zusätzlich eingeschränkt werden:

- a) **block\_rpc** – Der Aufruf von RPC-Funktionen ist nicht gestattet. Jedoch können Ausnahmen angegeben werden.
- b) **allow\_rpc** – Der Aufruf aller RPC-Funktionen ist gestattet.

Zusätzlich kann auch die Veröffentlichung von Daten eingeschränkt werden:

- a) **block\_write** – Veröffentlichung von Datenressourcen mittels der Methoden **announce** und **publish** blockiert (default)
- b) **allow\_write** – Veröffentlichung erlaubt

Die Regelsätze können vom Administrator des Systems, von installierten Anwendungen oder aus globalen Voreinstellungen vorgegeben werden.

Eine frühere Version dieser Regelsätze wurden in der Masterarbeit [16/019] näher untersucht.

### 5.1.6 Darstellung von Nutzerschnittstellen

Einige Funktionalitäten benötigen die Interaktion mit Nutzerinnen und Nutzern. Hierzu können zum Beispiel akustische oder visuelle Signale durch entsprechende Geräte und einer entsprechenden Geräteeinbindung mittels der IoT-Softwareplattform realisiert werden. Bildschirm-geführte Nutzerschnittstellen erfordern allerdings komplexere Schnittstellen. Aus diesem Grund wurde ein Framework implementiert, das den Anwendungen die Darstellung von Nutzerinterfaces ermöglicht. Dazu wurden die Kernmodule mit einer Bibliothek zur Definition von graphischen Nutzerinterface-Elementen nach einem Model-View-ViewModel-

Entwurfsmuster (knockout.js) zu einem separaten Framework kombiniert. Dieses kann in Webbrowsern, aber auch in sogenannten WebViews auf Smartphones und Desktop-Rechnern ausgeführt werden. Auch die Erstellung nativer Smartphone- oder Desktop-Anwendungen ist möglich. Über das in Unterabschnitt 5.1.4 definierte Netzwerkprotokoll kann eine solche Instanz auf die entfernten Datenressourcen anderer Instanzen der IoT-Softwareplattform zugreifen.

Anwendungen können Ansichten (Views) als HTML-Dateien und Ansichtsmodelle (ViewModels) als JavaScript-Dateien definieren. Diese müssen im Ordner „view“ im Verzeichnis der jeweiligen Anwendung abgelegt werden. Die Ansicht mit dem Namen „home“ entspricht der Ansicht des Eröffnungsfensters.

Die Konfiguration kann ähnlich zu der Konfiguration der IoT-Softwareplattform (siehe Einleitung zu Abschnitt 5.1) in einer Datei auf dem Webserver abgelegt werden. Auch eine Übergabe über die Adressleiste ist möglich.

## 5.1.7 Implementierte Anwendungen

Zur Realisierung von Geräteeinbindungen und Energiemanagement-Anwendungen wurden die nachfolgenden Hardware-Anbindungen realisiert:

Anbindung von verschiedenen Energiemesssteckdosen unterschiedlicher Hersteller:

- dlink-smartplug, plugwise

Anbindung von Bussystemen:

- i2c – I2C-Bus zur Kommunikation mit Sensoren in geringer Distanz (auf Computern mit I2C-Interface wie dem Raspberry Pi),
- gpio, gpio-out – GPIO-Pin: Manipulation und Auslesen (auf Mini-Computern wie dem Raspberry Pi),
- w1temp – 1Wire-Bus zur Kommunikation mit Temperatursensoren,
- knx – KNX-Bus zur Gebäudeautomation,
- modbus – Modbus zur Kommunikation mit Energiemesssystemen und Industriesteuerungen (siehe [Mod18]),
- ethercat – EtherCat zur Kommunikation mit Industriemesssystemen (siehe [JB04]),
- DMX usbdmx, ArtNet artnet und MIDI midi zur Kommunikation mit Lichtsteuerungssystemen.

Anbindung von Protokollen bzw. vorhandener Lösungen:

- **snmp-energy** – Auslesen von SNMP Attributen, wie dem Energieverbrauch von Servern,
- **mqtt** – MQTT-Protokoll zur Anbindung anderer IoT-Projekte (siehe Unterabschnitt 2.3.4),
- **coap** – CoAP-Protokoll zur Anbindung anderer IoT-Projekte (siehe Unterabschnitt 2.3.4),
- **rest-api** – Bereitstellung von Daten und RPC-Methoden von Datenressourcen als REST-API (siehe Unterabschnitt 2.3.4),
- **openhbab-bridge** – OpenHAB zur Kommunikation mit vorhandenen Smart-Home-Installationen,
- Diverse Hersteller-abhängige Systeme: **rcswitch**.

Weiterhin wurden folgende Hilfsanwendungen geschaffen:

Administration und Fehlersuche:

- **admin** – Administrationsinterface zur Änderung der Instanz-Konfigurationsdatei,
- **admin-app** – Administrationsinterface zur Konfiguration einzelner Anwendungen,
- **admin-update** / **install-apps-git** / **install-apps-npm** -- Routinen zur Installation und zum Update von IoT-Softwareplattform und Anwendungen,
- **web-admin** – Web-basiertes Administrationsinterface zum Überblick über die Datenressourcen,
- **ping-host** – Erfassung der Netzwerkumlaufzeit (engl. Round Trip Time, RRT) zwischen zwei Instanzen.

Vereinfachte Konfiguration:

- **template** – Definition und Wiederverwendung von Konfigurationsobjekten. Dabei können individuell Schlüssel ausgetauscht werden. Diese Anwendung vereinfacht wiederkehrende Konfigurations-Einträge,
- **config-include** – Einbindung weiterer Konfigurationsdateien.

Anbindung von externen Programmen:

- **execcommand** – Ausführung von Befehlen, wenn diese per RPC angesprochen werden,

- **childprocess** – Kontinuierliche Ausführung eines Programmes und Veröffentlichung der Rückgaben in einer Datenressource.

Zusätzlich wurden noch Anwendungen für die Überprüfung der Funktionen der IoT-Softwareplattform und deren Anwendungen erstellt. Im weiteren Verlauf des Kapitels 5 werden noch weitere Anwendungen erläutert.

### 5.1.8 Geräteeinbindung von vernetzten Geräten

Zur Realisierung von intelligenten Geräten (nach Abschnitt 4.1) und für die Einbindung von vernetzten Geräten wird auf die Datenressourcen der IoT-Softwareplattform zurückgegriffen. Der Funktionsumfang nach Unterabschnitt 4.1.1 wird folgendermaßen umgesetzt:

- Intelligente Geräte können durch eine DNS-basierte Dienstermittlung (engl. DNS Based Service Discovery, DNS-SD, siehe [CK13]) selbstständig den lokalen Koordinator im Netzwerk ermitteln. Nach der Herstellung der Verbindung werden automatisch die Datenressourcen im System verteilt, womit auch die **Existenz** der Geräte bekannt gegeben wird. Diese Dienstermittlung wird mit der Hilfsanwendung **mdns** umgesetzt. Ein ähnliches Verfahren über HTTP setzt die Anwendung **http-sd** um. Diese kann z. B. für Web- und Smartphone-Anwendungen eingesetzt werden, bei denen die DNS-basierte Ermittlung nicht möglich ist. Geräte, die über vorhandene Kommunikationslösungen eingebunden werden, benötigen spezielle Anwendungen und teilweise eine manuelle Konfiguration.
- Bei der Übermittlung der Datenressourcen durch die Basismethode **subscribe\_announcements** werden Metadaten mitübertragen, die die Datenressourcen und die enthaltenen Daten beschreiben. Diese Metadaten werden für die **Metadaten** der Geräte verwendet, die nach dem Funktionsumfang aus Unterabschnitt 4.1.1 gefordert werden.
- In dieser oder in weiteren Datenressourcen können zudem Informationen über den **Betriebsstatus** sowie **weitere Messdaten** verbreitet werden. Bei Bedarf können andere Anwendungen auf diese Daten mittels eines Aufrufes der Funktion **subscribe** zugreifen.
- Die optionale Funktionalität zur **Beeinflussung des Betriebsstatus** wird mit Hilfe einer weiteren Datenressource für die Bereitstellung der Prognose realisiert. Die Beeinflussung selbst wird über eine RPC-Funktion aufgerufen. Handelt es sich um ein IoT-Gerät auf dem eine Instanz der IoT-Softwareplattform verarbeitet wird, so können die Prognose-Berechnung und die RPC-Funktion direkt auf die Hardware zugreifen

und die gewünschten Beeinflussungen unmittelbar umsetzen. Wird hingegen über ein anderes Bussystem auf das Gerät zugegriffen, so ist es von dem Funktionsumfang des verwendeten Bussystems abhängig, ob die Prognose und die Beeinflussung direkt möglich sind. Andernfalls kann eine fehlende Beeinflussungsfunktion nach der Konzeption aus Unterabschnitt 4.1.6 nachgebildet werden. Die Beeinflussung selbst kann mittels einer Schaltfunktion, z. B. einer schaltbaren Zwischensteckdose nachempfunden werden. Anbindungen zu entsprechenden Produkten und Bussystemen wurden implementiert (siehe Unterabschnitt 5.1.7).

- Die geforderten **Erweiterungsmöglichkeiten** sind durch Software-Anwendungen (siehe Unterabschnitt 5.1.3) gegeben.

Im Rahmen dieser Arbeit wurden auf diese Weise erste reale IoT-Geräte auf Basis der IoT-Softwareplattform umgesetzt. Zusätzlich wurden für Testzwecke virtuelle IoT-Geräte implementiert, die den vollständigen oben genannten Funktionsumfang aufweisen, in der Realität jedoch nicht existieren. Weiterhin wurden zahlreiche Anbindungen von Bussystemen und Schnittstellen verwirklicht (siehe Unterabschnitt 5.1.7) und dadurch vernetzte Geräte in die IoT-Softwareplattform eingebunden.

### 5.1.9 Geräteeinbindung von nicht-vernetzten Geräten

Mit Hilfe des Konzepts der IoT-Geräterepräsentationen (siehe Unterabschnitt 4.2.1) können Geräte nachgebildet werden, deren Betriebsstatus nicht direkt vorliegt. Dieses Konzept wurde auf Basis der IoT-Softwareplattform in den nachfolgenden Anwendungen implementiert. Dabei wurden die in Unterabschnitt 5.1.8 genannten Lösungsmöglichkeiten angewendet. Die realisierten Anwendungen können direkt zur Geräteeinbindung oder indirekt von anderen Anwendungen mittels der Vererbung von Funktionen verwendet werden.

Die Anwendung **device/iot-representation** wurde als Basisanwendung erstellt, von der andere Anwendungen Funktionalitäten erben können. Sie gibt Strukturen vor, implementiert jedoch keine Funktionalität. Für die Nachbildung des Betriebsstatus wird eine Funktion für deren Initialisierung bereitgestellt, die dann von den jeweiligen Anwendungen überschrieben werden muss. Zusätzlich werden Datenressourcen für den Betriebsstatus, die Gerätebeeinflussung und den Energieverbrauch erzeugt. Auch für diese Daten stehen Funktionen bereit, die überschrieben werden können. Wie in der Kapiteleinführung zu Kapitel 5 benannt, werden die Umsetzungen zur Zustandsidentifikation im späteren Verlauf erläutert.

Durchgängig betriebene, nicht geregelte Geräte, die ihren Betriebsstatus und damit auch ihr Verhalten langfristig nicht verändern, können mit dieser Anwendung direkt eingebunden wer-



den, da alle notwendigen Daten bekannt sind oder sich mit wenig Aufwand ermitteln lassen. Ein Beispiel ist die Anwendung device/state-static für Geräte, die ihren Betriebsstatus nicht verändern. Sie kann direkt verwendet oder für die Vererbung genutzt werden. Für den Betriebsstatus wird dazu einmalig der Status „An“ in der entsprechenden Datenressource veröffentlicht. Eine Beeinflussung des Status ist nicht vorgesehen. Zusätzlich kann auch der Energieverbrauch mit veröffentlicht werden, der im Sinne der semi-statischen Geräte aus Abschnitt 4.2 als konstant vorausgesetzt wird. Dazu wird aus der Konfiguration der Anwendung ein einzelner fester Wert für die Leistungsaufnahme entnommen und in der entsprechenden Datenressource veröffentlicht. Dies kann beispielsweise auf eine nicht geregelte und durchgängig betriebene Heizungspumpe angewendet werden. Der Energieverbrauch kann durch die auf dem Typenschild angegebene Leistungsaufnahme hochgerechnet werden.

Auch wenn die Heizungspumpe nach einem festen Zeitablauf gesteuert wird, ist eine Hochrechnung auf diese Art möglich. Dazu muss der Anwendung der Zeitplan bei der Initialisierung bekannt gegeben werden. Exemplarisch wird hier auf die Anwendung device/state-schedule verwiesen. Sie kann ebenfalls neben der direkten Nutzung auch für die Vererbung verwendet werden. Diese Anwendung erbt selbst die Methode der Energieverbrauchshochrechnung der Anwendung device/state-static.

In der Realität sind solche gut prognostizierbaren Geräte nur vereinzelt anzutreffen. Deshalb werden im nächsten Abschnitt Techniken zur Identifikation des Status anhand von Messdaten vorgestellt.

## 5.2 Signalverarbeitung für die Geräteeinbindung

In diesem Abschnitt werden aus dem Bereich der Signalverarbeitung bekannte Verfahren eingesetzt, um die Geräteeinbindungen mit weiteren Informationen anzureichern.

Die Geräteeinbindung von **vernetzten Geräten** (nach der Definition aus der Einleitung zu Kapitel 4) anhand der IoT-Softwareplattform wurde bereits in Unterabschnitt 5.1.8 erläutert. Teilweise fehlen bei diesen vernetzten Geräten Messdaten, die z. B. für das Energiemanagement von Bedeutung sein können. Dazu wurde in Unterabschnitt 4.1.4 eine Technik zur Nachbildung fehlender Informationen vorgeschlagen. In Unterabschnitt 5.2.2 wird diese Technik beispielhaft für die Nachbildung des Energieverbrauchs vorgestellt und in Unterabschnitt 5.2.3 mittels einer direkten Zuordnung umgesetzt. In Unterabschnitt 5.2.4 wird eine lineare Regression, aber auch ein neuronales Netz für die Zuordnung verwendet.

Im Kontext dieser Arbeit wurden auch erste intelligente Geräte auf Basis der Softplattform als Demonstratoren entwickelt. Durch die Anwendung der Verfahren zur Nachbildung fehlender

Informationen, lassen sich bei diesen Geräten ggfs. Kosten durch Wegfall zusätzlicher Sensoren einsparen.

**Nicht-vernetzte Geräte** können ihren Betriebsstatus nicht direkt bereitstellen. Wie in Unterabschnitt 5.1.9 erläutert, soll hierfür die Einbindung im Sinne einer IoT-Geräterepräsentation verwendet werden. Zusätzlich zu der Möglichkeit der Integration semi-statischer Geräte, die bereits ebendort erläutert wurde, wurden in Abschnitt 4.2 noch drei weitere Verfahren behandelt, zu denen in diesem Abschnitt verschiedene Implementierungen vorgestellt werden. Bei diesen Verfahren soll während des Gerätebetriebes mittels zusätzlicher geeigneter Sensoren oder Daten auf den Betriebsstatus geschlossen werden.

In Unterabschnitt 5.2.5 wird dazu zunächst ein Überblick über die verwendeten Messdaten und die implementierten Algorithmen gegeben. Darauf aufbauend wird in Unterabschnitt 5.2.6 eine Technik vorgestellt, die eine Klassifikation anhand von Schwellenwerten durchführt. Für diese Klassifikation wird in Unterabschnitt 5.2.7 eine Parametrisierung durchgeführt, die die Abbildung möglichst vieler Geräte ohne weitere Adaption erlauben soll. In Unterabschnitt 5.2.8 wird eine Technik vorgestellt, die durch einen Musterabgleich mittels des Algorithmus zur dynamischen Zeitanpassung (engl. Dynamic Time Warping, DTW) automatisch Programmabläufe identifiziert. In Unterabschnitt 5.2.9 wird die Zustandsidentifikation mehrerer Geräte anhand von Neuronalen Netzen unter Verwendung von Messdaten der Netzspannungsbeeinflussungen des lokalen 230-V-Versorgungsnetz erläutert.

Automatische Trainingsverfahren der Mustererkennung, wie Hidden-Markoff-Modelle oder künstliche neuronale Netze, erfreuen sich großer Beliebtheit in der Signalverarbeitung. Aus diesem Grund werden solche Verfahren auch in einigen der benannten Techniken eingesetzt. Viele Trainingsverfahren erfordern jedoch die Verfügbarkeit einer hohen Anzahl an klassifizierten Trainingsdaten. Deshalb wurde in dieser Arbeit zunächst auf vergleichsweise einfache Verfahren zurückgegriffen, die mit möglichst wenigen oder sogar auch ohne Trainingsdaten auskommen.

Die Implementierungen der benannten Erkennungs- und Nachbildungsverfahren wurden jeweils in der Skriptsprache der numerischen Berechnungsumgebung MATLAB oder bzw. und der Skriptsprache JavaScript umgesetzt. Dabei wurde besonders auf eine Online-Fähigkeit (direkte Datenverarbeitung) der Umsetzungen geachtet, damit diese auch in der Praxis problemlos genutzt werden können. Um die neuen Messdaten fortlaufend verarbeiten zu können, ist eine permanente Funktionsbereitschaft der Erkennungs- und Nachbildungsverfahren zwingend erforderlich. Dazu wurde im Regelfall für alle Techniken eine spezifische Implementierungsstruktur gewählt. Diese wird nachfolgend in Unterabschnitt 5.2.1 erläutert.

### 5.2.1 Direkte Datenverarbeitung

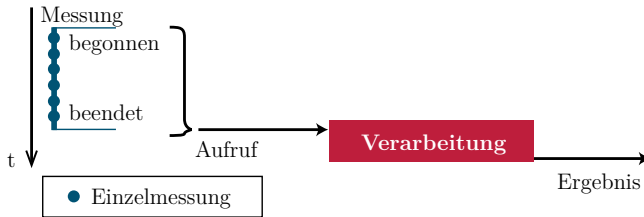


Abbildung 5.3: Struktur bei abschließender Verarbeitung. Zunächst werden alle Messdaten gesammelt und nach Ende der Messung zusammen der Verarbeitung übergeben.

Viele artverwandte Anwendungsgebiete der Erkennungs- und Nachbildungsverfahren können den Vorteil nutzen, dass ihr Dienst nur bei Bedarf benötigt wird. Beispielsweise wird die Spracherkennung in der Regel nur für kurze Zeit nach einer Aktivierung verwendet. In der überwiegenden Anzahl der Fälle muss die Verarbeitung zudem nicht direkt (live) erfolgen, sondern es genügt eine zeitnahe Verarbeitung nach der vollständigen Aufnahme der Eingangsdaten. Diese Verarbeitungsstruktur ist in Abbildung 5.3 visualisiert.

Die beiden genannten Vereinfachungen können für die meisten der nachfolgenden Erkennungs- und Nachbildungsverfahren nicht angenommen werden. Dies liegt u. a. daran, dass eine Aktivität eines Gerätes jederzeit erfolgen kann und die mögliche Einschaltdauer im Vorherin in der Regel nicht bekannt ist. Eine Dauer der Geräteaktivität bei Geräten in Privathaushalten von mehreren Stunden ist keine Seltenheit. Energiemanagement-Lösungen mit Nutzereinbindung sowie auch automatisierte Energiemanagement-Lösungen erfordern jedoch im Vergleich zu der jeweiligen Aktivitätsdauer kürzere Zeiten für deren Rückmeldung und damit auch für den Prozess der Erkennung bzw. Verarbeitung (siehe Unterabschnitt 2.1.2).

Aufbauend auf den Software-Anwendungen (siehe Unterabschnitt 5.1.3) und den Datenressourcen (siehe Unterabschnitt 5.1.1 und 5.1.2) wurde die Verarbeitungsstruktur für eine andauernde direkte Verarbeitung in Abbildung 5.4 konzipiert. Mit dem Begriff der andauernden Verarbeitung wird das verwendete Prinzip beschrieben, bei dem die Verarbeitung nicht auf Anfrage, sondern in regelmäßigen Abständen in der Regel nach jedem Messzyklus erfolgt. Die regelmäßig aufgenommenen Messdaten werden hierzu in der Regel durch eine eigenständige Software-Anwendung der IoT-Softwareplattform in einer Datenressource veröffentlicht. Die gewünschte Verarbeitung in Form der Erkennungs- und Nachbildungsverfahren wird ebenfalls als Software-Anwendungen der IoT-Softwareplattform implementiert. Nach ihrer Initialisierung können diese die angebotenen Datenressourcen abonnieren. Dabei geben

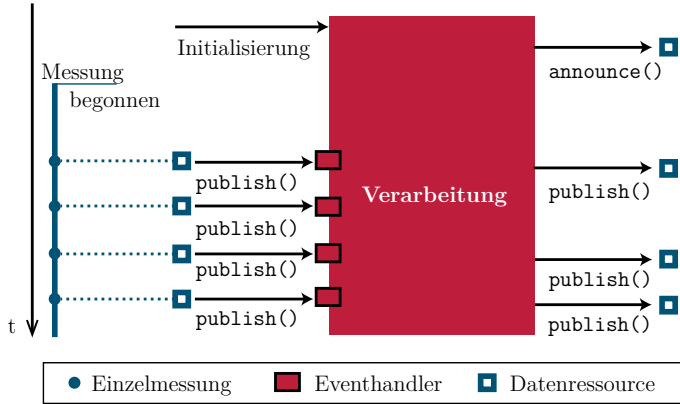


Abbildung 5.4: Struktur bei andauernder direkter Verarbeitung. Die wiederkehrend aufgenommenen Messdaten werden in einer Datenressource veröffentlicht, die von den implementierten Erkennungs- und Nachbildungsverfahren in Form von Software-Anwendungen abonniert werden können. Die Ergebnisse werden in einer weiteren Datenressource veröffentlicht.

sie einen Callback an, der anschließend wiederkehrend aufgerufen wird und die Messdaten als Argument erhält. Dadurch ist eine direkte Verarbeitung mit geringer Verzögerung möglich. Die erzeugten Ergebnisse werden anschließend in einer weiteren Datenressource veröffentlicht.

### 5.2.2 Energieverbrauchsermittlung aus Statusdaten

Wie in Unterabschnitt 4.1.4 vorgeschlagen, sollen fehlende Daten aus dem Betriebsstatus und anderen Statusdaten nachgebildet werden können. Das zentrale Element dieses Konzepts ist eine Abbildung der Statusdaten in die zu schätzenden Daten. In Abhängigkeit von den bereitstehenden Daten und den zulässigen Annahmen über die Geräte ist die Verwendung von jeweils anderen Abbildungsverfahren, wie z. B. Transformationen, Zuordnungsalgorithmen bzw. Schätzern sinnvoll. Hierzu werden nachfolgend in Unterabschnitt 5.2.2 und 5.2.3 unter verschiedenen getroffenen Annahmen unterschiedliche teilweise vereinfachte Varianten dargestellt.

Alle Varianten setzen die Aufgabe der Ermittlung des Energieverbrauchs um. Der für den jeweiligen Einsatzzweck konkret ermittelte Energieverbrauch eines Gerätes ist sowohl für die Nutzerinnen und Nutzer, um eigenständig geeignete Maßnahmen zur Reduzierung ihres Energieverbrauch zu identifizieren, aber auch für die Algorithmen des automatisierten

Energiemanagements von Interesse. Die überwiegende Anzahl der Implementierungen eignet sich auch für die Schätzung von anderen Größen.

Die benötigten Statusdaten können aus unterschiedlichen Quellen stammen. Beispielsweise genügt in der Regel der Betriebsstatus eines intelligenten Gerätes, wie nach Unterabschnitt 4.1.1 definiert. Aber auch sonstige Daten, die die Aktivitäten eines Geräts ausreichend beschreiben, können genutzt werden. Auch der geschätzte Zustand aus einer Zustandsidentifikation nach Abschnitt 4.2 kann verwendet werden.

### 5.2.3 Direkte Zuordnung des Energieverbrauchs

Die einfachste Variante ist die direkte Abbildung vom Betriebsstatus  $s(t) = S_j$ , mit

$$P(t) = \begin{cases} P_j, & \text{wenn } s(t) = S_j, \quad \forall j \in \{1, \dots, J\} \\ P(t-1), & \text{sonst} \end{cases},$$

auf feste korrespondierende Werte  $P_j$ . Hier wird angenommen, dass der Betriebsstatus höchstens  $J$  verschiedene Ausprägungen annimmt. Dennoch sind auch Intervalldefinitionen möglich.

Bei vielen Geräten sind für die Abbildung lediglich zwei Ausprägungen erforderlich: Ein nicht-produktiver Zustand „Aus“ mit  $s_1 = \text{“off“}$  und ein produktiver Zustand „An“ mit  $s_2 = \text{“on“}$ . Für den produktiven Zustand genügt in der Regel ein einzelner fester Wert für die Leistungsaufnahme. Für den nicht-produktiven Zustand müssen Standby-Verbräuche und Restströme beachtet werden. Treten diese nicht auf oder sollen diese unter Inkaufnahme der Folgefehler vernachlässigt werden, kann die Leistungsaufnahme zu  $P_1 = 0$  Watt gewählt werden. Ist beispielsweise der Betriebsstatus einer Leuchte bekannt, die mit einer Leuchtstoffröhre mit einer Nennleistung von 58 Watt betrieben wird, so lautet die Abbildung: „Aus“:  $P_1 = 0$  Watt, „An“:  $P_2 = 58$  Watt. Für komplexere Geräte können mehrere produktive Zustände definiert werden, jedoch können mit dieser Abbildung Verläufe und komplexe Programmabläufe nur bedingt abgebildet werden.

Die Qualität dieser Abbildung ist stark abhängig von den verwendeten Daten für die Leistungsaufnahmen  $P_j$ . Es empfiehlt sich, diese Werte durch eine einmalige manuelle Messung zu kontrollieren. Weitere Effekte, die die Leistungsaufnahme beeinflussen, wie thermische Effekte (z. B. bei Leuchtmitteln), Alterungseffekte oder Abhängigkeiten zur Umgebung, können mit dieser vereinfachten Variante nur sehr bedingt abgebildet werden.

Diese Energieverbrauchsschätzung ist in der Anwendung device/state-static implementiert, die bereits in Unterabschnitt 5.1.9 erwähnt wurde und kann auch auf nicht statische

Geräte angewendet werden.

Nachfolgend werden zwei weitere Beispiele für die Energieverbrauchsermittlung beschrieben.

### 5.2.4 Energieverbrauchsermittlung von Desktop-Computern

Die Leistungsaufnahme von Computern wird im Wesentlichen von der jeweiligen Rechenleistung bestimmt. Im Gegensatz zu Laptops besitzen Desktop-Computer in der Regel keinen Sensor für die Ermittlung ihres Energieverbrauchs. Dennoch sind mit geeigneter Software auf einem Desktop-Computer viele Informationen über die Hardware und deren Nutzung zugänglich.

Ebenfalls auf der Grundlage von Unterabschnitt 4.1.4 sollen solche Daten im Folgenden für die Energieverbrauchsschätzung genutzt werden. In Voruntersuchungen wurde dabei festgestellt, dass von den untersuchten Informationsquellen insbesondere die Nutzungsdaten des Hauptprozessors (CPU) und des Grafikprozessors (GPU) entscheidende Abhängigkeit zur Leistungsaufnahme aufwiesen. Alle anderen ausgewerteten Größen waren entweder von diesen beiden Größen abhängig oder besaßen innerhalb der Messungenauigkeit der verwendeten Messgeräte keinen Anteil. Daraus ergibt sich unter Verwendung der Abbildung M die geschätzte Leistungsaufnahme eines eingeschalteten Computers zu

$$P(t) = M_{\text{Computer}}(L_{\text{CPU}}(t), L_{\text{GPU}}(t)),$$

wobei  $L_{\text{CPU}}(t)$  und  $L_{\text{GPU}}(t)$  der Nutzung von CPU und GPU entsprechen und in Prozent zwischen 0 % und 100 % angegeben sind. Die Abbildung M ist dabei individuell für jeden Computer zu wählen bzw. zu bestimmen. Es kann jedoch angenommen werden, dass die Abbildung zumindest auch auf Computer der gleichen Baureihe übertragen werden kann.

Für die Realisierung der Abbildung M gibt es viele Möglichkeiten. Unter der Annahme, dass die Komponenten unabhängig sind, ergibt sich die geschätzte Leistungsaufnahme zu

$$P_I(t) = M_{\text{CPU}}(L_{\text{CPU}}(t)) + M_{\text{GPU}}(L_{\text{GPU}}(t)).$$

Die Abbildungen  $M_{\text{CPU}}$  und  $M_{\text{GPU}}$  sind ebenfalls individuell für jeden Computer bzw. jede Baureihe zu wählen bzw. zu bestimmen. Der Verbrauch im Leerlauf muss in einer der beiden Abbildungen enthalten sein.

Für die lineare Näherung wurde die Leistungsaufnahme als

$$P_{\text{II}}(t) = P_{\text{Grundlast}} + P_{\text{CPU}} \cdot L_{\text{CPU}}(t) + P_{\text{GPU}} \cdot L_{\text{GPU}}(t)$$

angenommen, wobei  $P_{\text{Grundlast}}$ ,  $P_{\text{CPU}}$  und  $P_{\text{GPU}}$  die Leistungsaufnahme im Leerlauf sowie die maximalen Leistungsaufnahmen der CPU und der GPU darstellen.

Die Leistungsaufnahmen müssen mit separaten Messungen ermittelt werden. Man führt dazu die folgenden drei Messungen unter Verwendung von Softwarewerkzeugen zur Erzeugung von CPU- bzw. GPU-Lastwerten durch:

- Leerlauf:  $P \rightarrow P_{\text{Grundlast}}$  mit  $L_{\text{CPU}}(t) \approx 0\%$  und  $L_{\text{GPU}}(t) \approx 0\%$
- Volle Rechenlast:  $P \rightarrow P_{\text{Grundlast}} + P_{\text{CPU}}$  mit  $L_{\text{CPU}}(t) \approx 100\%$  und  $L_{\text{GPU}}(t) \approx 0\%$
- Volllast:  $P \rightarrow P_{\text{Grundlast}} + P_{\text{CPU}} + P_{\text{GPU}}$  mit  $L_{\text{CPU}}(t) \approx 100\%$  und  $L_{\text{GPU}}(t) \approx 100\%$

Aus diesem linearen Gleichungssystem können die Koeffizienten  $P_{\text{Grundlast}}$ ,  $P_{\text{CPU}}$  und  $P_{\text{GPU}}$  bestimmt werden. Prinzipiell könnte auch eine vierte Messung mit voller GPU-Last und sehr geringer CPU-Last für die Bestimmung verwendet werden. Dieser Zustand ist technisch jedoch nur mit großem Aufwand realisierbar, da die Erzeugung von GPU-Last auch eine entsprechende CPU-Last verursacht. Aus diesem Grund wurde auf die Hinzunahme dieser Messung verzichtet.

Für den Live-Betrieb wurde eine Software-Anwendung **pc-energy-estimation** für die Software-Plattform implementiert. Dazu wurden die oben benannten Gleichungen für  $P_I(t)$  und  $P_{\text{II}}(t)$  implementiert. Für das Training der Abbildungen wurde auch die zeitgleiche Aufnahme des Energieverbrauchs und der Nutzungsgrößen über die IoT-Softwareplattform implementiert. Dadurch können die Koeffizienten  $P_{\text{Grundlast}}$ ,  $P_{\text{CPU}}$  und  $P_{\text{GPU}}$  sowie auch die Abbildungen  $M_{\text{CPU}}$  bzw.  $M_{\text{GPU}}$  automatisiert bestimmt werden.

Eine Schätzung des Energieverbrauchs für Server ist bereits aus [LGT08] bekannt. Hier wurden andere Computer-Komponenten fokussiert und eine andere Art der Modellierung gewählt. In der Bachelorarbeit [15/704] wurde auch die Anwendung eines neuronalen Netzes für die Abbildung  $M$  implementiert und erprobt. Dies hat den Vorteil, dass mit wenigen Messungen auch komplexe Abbildungen erzeugt werden können. Diese Arbeit fokussiert jedoch auf die grundsätzliche Anwendbarkeit der Technik, weshalb die Verwendung des neuronalen Netzes hier nicht weiter untersucht wird.

### 5.2.5 Zustandsidentifikation aus zusätzlichen Messdaten

In Abschnitt 4.2 wurden verschiedene Verfahren zur Einbindung von nicht-vernetzten Geräten benannt. Alle Verfahren fokussieren auf der Identifikation des Betriebsstatus eines oder mehrerer Geräte. In den nachfolgenden Unterabschnitten werden die Implementierungen dieser Verfahren vorgestellt, die im Rahmen dieser Arbeit umgesetzt wurden. Aus den in Unterabschnitt 4.2.2 benannten Sensoren wurden dazu Energieverbrauchs- und Helligkeitssensoren sowie Sensoren für Netzspannungsbeeinflussungen (Elektromagnetische Beeinflussungen) verwendet. Andere Sensoren könnten in der Regel jedoch ebenfalls Anwendung finden. Für die drei in Abschnitt 4.2 benannten Erkennungsaufgaben der *Zustandsidentifikation von einzelnen Geräten*, der *Zustandsidentifikation mehrerer Geräte* und der *Identifikation des Programmablaufs* wurden die folgenden Algorithmen entwickelt:

In Unterabschnitt 5.2.6 wird zunächst die regelbasierte Erkennung anhand einer Schwellenwert-Klassifikation erläutert. Diese Erkennung wird im Rahmen dieser Arbeit unter Verwendung von Energieverbrauchs- und Helligkeitssensoren zur Zustandsidentifikation von einzelnen Geräten eingesetzt. Für dieses Einbindungsverfahren wird in Unterabschnitt 5.2.7 ein Modell vorgestellt, das für die Klassifikation möglichst vieler Gerätklassen eingesetzt werden soll.

Künstliche neuronale Netze werden in Unterabschnitt 5.2.9 in Kombination mit Sensoren für Netzspannungsbeeinflussungen für die Zustandsidentifikation mehrerer Geräte vorgestellt.

Maschinelles Lernen erfordert in der Regel große Mengen klassifizierter Trainingsdaten. In Unterabschnitt 5.2.8 wird eine Umsetzung zum Musterabgleich aufbauend auf dem Verfahren der dynamischen Zeitanpassung erläutert. Dieses Einbindungsverfahren soll mit nur wenigen antrainierten Mustern verwendet werden können.

In Ergänzung zu den beschriebenen Kombinationen wäre für diese Arbeit auch eine Zustandsidentifikation mehrerer Geräte anhand von akkumulierten Energieverbrauchsverläufen unter Verwendung von maschinellem Lernen interessant gewesen. Hierzu wurden in der Masterarbeit [13/008] eine Geräteerkennung basierend auf Hidden-Markoff-Modellen entwickelt. Jedoch benötigt das maschinelle Lernen große Mengen an klassifizierten Trainingsdaten mit den tatsächlich-angenommenen Betriebszuständen. Da diese Daten nicht in der gewünschten Menge und Qualität zur Verfügung standen, konnte die Entwicklung und Evaluation dieser Technik nicht abgeschlossen werden. An dieser Stelle sei deshalb auf andere Forschungsgruppen verwiesen, die sich dieser Erkennungsaufgabe gewidmet haben (siehe Unterabschnitt 2.2.3 und z. B. [WLR18]). Die Verwendung dieser Techniken kann aber grundsätzlich ähnlich umgesetzt werden, wie bei der Mehrgeräteerkennung auf Basis von Netzspannungsbeeinflussungen gezeigt wird.



### 5.2.6 Zustandsidentifikation mittels Schwellenwert-Klassifikation

Diese Implementierung basiert auf dem Konzept der regelbasierten Zustandsidentifikation aus Unterabschnitt 4.2.3 und wurde in der Skriptsprache JavaScript als Softwareanwendung **analysis/eventdetection** für den Live-Betrieb direkt in der IoT-Softwareplattform realisiert. Die Transformation zur Merkmalsextraktion ist nicht Teil dieser Anwendung. Sie kann aber als separate Anwendung zur Vorverarbeitung der Sensordaten vorgeschaltet werden. Dazu wurden beispielsweise in der Anwendung **analysis** verschiedene Verfahren implementiert.

Zusätzlich zur Definition des Erkenners in Unterabschnitt 4.2.3 für die Zustandsfolge  $s(t)$ , kann für die Erkennung zudem eine Mindest-Verbleib-Zeit  $D_j$  individuell für jedes Intervall  $j$  definiert werden. Ein neuer Zustand  $s^*(t)$  wird mit

$$s^*(t) = \begin{cases} s(t), & D_j == 0 \\ s(t), & s(t - \delta) == s(t), \forall \delta \in \{1, \dots, D_j\} \\ s^*(t - 1), & \text{sonst} \end{cases}$$

nur übernommen, wenn der Zustand auch zu den vorherigen diskreten Zeitpunkten  $s(t - \delta)$  mit  $\delta \in \{1, \dots, D_j\}$  erkannt wurde. Da ansonsten der vorherige Zustand  $s^*(t - 1)$  beibehalten wird, können hiermit kurze Peaks ignoriert werden. Um den neuen Zustand anzunehmen, darf innerhalb der Mindest-Verbleibzeit  $D_j$  für keinen Zeitschlitz ein anderer Zustand erkannt werden. Ist die Mindest-Verbleib-Zeit  $D_j$  zu Null gewählt, wird der erkannte Zustand  $s(t)$  direkt übernommen.

Diese Schwellenwert-Klassifikation kann für viele verschiedene Einsatzzwecke angewendet werden. Im Rahmen dieser Arbeit wurde diese in Kombination mit Energiemessgeräten und Helligkeitssensoren eingesetzt.

Die notwendigen Parameter für diese Klassifikation werden als sogenanntes Modell zusammengefasst (siehe auch Unterabschnitt 4.2.3). Dieses Modell ist der Anwendung **analysis/eventdetection** als Konfiguration zu übergeben. Bei der manuellen Parametrisierung eines Modells müssen für jeden zu erkennenden Zustand die nachfolgenden Daten angegeben werden. Zur Definition mehrerer Intervalle kann ein Zustand auch mehrmals auftreten:

- **name** (String): Name des Zustands  $S_j$ .
- **from, to** (Number): Untere und obere Intervallgrenzen  $o_j, u_j$ . Standardwerte sind minus bzw. plus unendlich.
- **mincount** (Number): Mindest-Verbleib-Zeit  $D_j$ . Als Zeiteinheit wird hier die Samplelänge zugrunde gelegt. Fehlende Sample werden nicht mitgezählt.

- Weitere beschreibende Daten, z. B. Kennzeichnung, ob es sich um einen produktiven Zustand handelt: `active` (Boolean)

### 5.2.7 Modell für die Zustandsidentifikation mittels Schwellenwert-Klassifikation

Für die gerade eingeführte Schwellenwert-Klassifikation (siehe Unterabschnitt 5.2.6) wurde ein Modell ermittelt, welches den Betriebsstatus für viele Gerätetypen anhand des elektrischen Energieverbrauchs  $P(t)$  ermitteln soll. Es wurde keine vorgeschaltete Transformation verwendet. Als Messstellen sollen Zwischenstecker mit Energiemessfunktion zum Einsatz kommen, weshalb die Parametrisierung nur für Niederspannungsanwendungen bis 230 V Betriebsspannung und einem maximalen Laststrom von 16 A durchgeführt wurde. Die genutzten Regelsätze wurden experimentell in einer Büro-Umgebung ermittelt in der auch einzelne Haushaltsgeräte zum Einsatz kamen. Um verfälschte Auswertungsergebnisse zu vermeiden, wurde diese Umgebung mit ihren Geräten nicht für die Evaluation verwendet. Damit wurde eine Separation in unabhängige Trainings- und Testdaten sichergestellt.

| Bedingung   | Zustand „Aus“   | Zustand „An“  |
|---|---|---|
| <b>Normal</b><br>$P_{\max} < 200 \text{ Watt}$<br>$P_{\max} \geq 10 \text{ Watt}$ | $o_j: P(t) \leq 5 \text{ Watt}$<br>$D_j = 2 \text{ S}$                          | $u_j: P(t) \geq 20 \text{ Watt}$<br>$D_j = 2 \text{ S}$   |
| <b>Große Last</b><br>$P_{\max} \geq 200 \text{ Watt}$                             | $o_j: P(t) \leq 10 \text{ Watt}$<br>$D_j = 2 \text{ S}$                         | $u_j: P(t) \geq 50 \text{ Watt}$<br>$D_j = 2 \text{ S}$   |
| <b>Geringe Last</b><br>$P_{\max} < 10 \text{ Watt}$<br>$P := P_{\text{avg}}$      | $o_j: P_{\text{avg}}(t) \geq 1,5 \text{ Watt}$<br>$D_j = 2 \text{ S} \cdot T_m$ | $u_j: P_{\text{avg}}(t) \geq \frac{P_{\min} + P_{\max} - P_{\min}}{2} (*)$<br>$D_j = 2 \text{ S} \cdot T_m$ |

Abbildung 5.5: Parameter des Modells. S steht hier für eine Samplelänge.  $T_m$  ist die Intervallgröße der Mittelwertbildung.

(\*) gilt nur bei  $P_{\max} > 2,5 \text{ Watt}$  und  $P_{\max} - P_{\min} > 3 \text{ Watt}$ .

Das Modell besteht aus drei Regelsätzen, die adaptiv anhand der bis dahin ermittelten maximalen Leistungsaufnahme  $P_{\max}$  des Gerätes gewählt werden. Die minimale und maximale Leistungsaufnahmen  $P_{\min}$  und  $P_{\max}$  werden nach einer laufenden Mittelwertbildung der letzten sechs Sekunden bestimmt, um kurzzeitige Schwankungen der Einzelwerte auszugleichen. Die gewählte Zeitspanne stellt einen erprobten Kompromiss zwischen einer akzeptablen

Verarbeitungsgeschwindigkeit und einer ausreichend genauen Präzision dar. Die folgenden Parameter sind zur Übersicht nochmal in Abbildung 5.5 aufgeführt.

- Die Erkennung startet im Regelsatz **Normal**. Der produktive Zustand „An“ wird ab einer Leistungsaufnahme von 20 Watt erkannt. Eine Obergrenze für die Leistungsaufnahme wurde nicht festgelegt. Der Zustand „Aus“, der auch den Standby-Betrieb der Geräte mit abdeckt, wird in einem Intervall von null bis fünf Watt erkannt. Die Mindest-Verbleibzeit beträgt für alle Zustände zwei Samplelängen.
- Da Geräte mit **größerer Leistungsaufnahme** teilweise auch größere Standby-Verbräuche aufweisen, wurden hierfür zusätzliche Intervalle bestimmt. Diese Intervalle werden genutzt, sobald das betreffende Gerät erstmals eine Leistungsaufnahme von größer als 200 Watt aufweist. Der Zustand „An“ wird nun ab einer Leistungsaufnahme von 50 Watt und der Zustand „Aus“ im Intervall von null bis zehn Watt erkannt.
- Auch für **geringe Lasten** wurde ein Regelsatz hinzugefügt. Viele der verwendeten Messgeräte wiesen bei einer sekundlichen Abtastung nur eine Wertauflösung von bis zu zwei Watt auf. Da die Mittelung über mehrere Werte bei diesen Messgeräten eine wesentlich höhere Auflösung gestattet, wurde zusätzlich eine laufende Mittelwertbildung  $P_{\text{avg}}(t)$  über die vergangenen  $T_m = 6$  Sekunden eingefügt. Diese wird aktiviert, sobald die maximale Leistungsaufnahme des betreffenden Gerätes unter einen Wert von neun Watt sinkt. Die obere Schwelle für die Leistungsaufnahme des Zustands „Aus“ wird zusätzlich auf 1,5 Watt gesenkt. Um die Mittelwertbildung auszugleichen wird weiterhin die Mindest-Verbleibzeit um den Faktor der Intervallgröße der Mittelwertbildung  $T$  erweitert. Auch die untere Schwelle für die Leistungsaufnahme des Zustands „An“ wird auf den Wert  $u_{j, \text{An, geringe Last}} = P_{\text{min}} + \frac{P_{\text{max}} - P_{\text{min}}}{2}$  angepasst, jedoch nur dann, wenn die maximale Leistungsaufnahme über 2,5 Watt und die Differenz zwischen minimaler und maximaler Leistungsaufnahme über 3 Watt liegt. Die Prüfung dieser Bedingungen soll verhindern, dass nicht-produktive Zustände als produktive Zustände erkannt werden.

Das Modell wurde als weitere Software-Anwendung **analysis/eventdetection-auto** implementiert. Diese Anwendung erbt die Funktionen der Software-Anwendung für die Schwellenwert-Klassifikation. Lediglich die Funktion zur Zuordnung der Modellparameter wurde überschrieben.

### 5.2.8 Zustandsidentifikation anhand des DTW-Musterabgleichs

In der Praxis werden zunächst vor allem Geräte im Einsatz sein, zu denen (vor Ort) keine Trainingsdaten existieren. Weiterhin kann es teilweise sehr aufwendig sein, Trainingsdaten in gewünschter Menge vor Ort aufzunehmen. Dies betrifft insbesondere Geräte mit komplexen, festen Programmabläufen, wie sie in Waschmaschinen, Spülmaschinen und im Besonderen bei Industriemaschinen zu finden sind. Diese Geräte sind aufgrund ihres Programmverhaltens mutmaßlich nur bedingt mit den genannten Regel-basierten Verfahren abzubilden. Die Verwendung von automatischen Trainingsverfahren ist deshalb gerade bei solchen Geräten interessant.

Bei Programmabläufen handelt es sich in der Regel um zeitlich ausgedehnte wiederkehrende Abfolgen. Bei vielen dieser Geräte spiegeln sich die Abläufe auch im zeitlichen Verlauf des Energieverbrauchs der Geräte wider. Im Rahmen dieser Arbeit wurden aus der oben genannten Argumentation heraus zunächst Verfahren aus dem Bereich der Mustererkennung betrachtet, die sehr wenige Trainingsdaten erfordern.

Das Verfahren der *dynamischen Zeitanpassung* (engl. *Dynamic Time Warping*, DTW) wurde in den 1970er Jahren unter anderem für die ersten Versuche im Bereich der Spracherkennung verwendet. Bei diesem Verfahren werden zwei Signale Sample pro Sample miteinander verglichen. Durch die Wiederholung von Samplewerten durch den Algorithmus werden dabei die beiden Zeitachsen der Signale so gedehnt, dass die Signalformen optimal übereinander passen. Der Grad der Übereinstimmung wird dabei mit Hilfe eines Distanzmaßes berechnet. Das Distanzmaß gibt schließlich auch die Qualität der Übereinstimmung der beiden Signale wieder. (vgl. [PK08, S. 312ff])

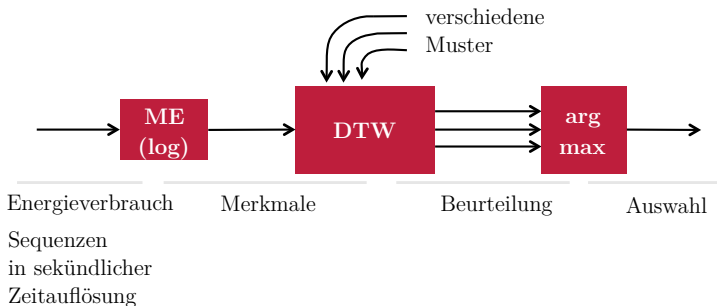


Abbildung 5.6: Ablauf des Abgleichs der Leistungsaufnahme mit verschiedenen Mustern, bestehend aus einer Merkmalsextraktion (ME), dem DTW-Musterabgleich und einer Auswahl des Musters mit der geringsten Abweichung.

In Analogie zur Struktur in Unterabschnitt 4.2.4 ist in Abbildung 5.6 die Struktur zur Erkennung mittels des DTW dargestellt. Als Messdaten werden Sequenzen der Leistungsaufnahme mit einer sekundlichen zeitlichen Auflösung verwendet. Für die Merkmalsextraktion wird eine Logarithmierung der Eingangswerte angewendet. Dieses bewirkt eine Dynamikkompression der Eingangswerte, wodurch die kleinen Signalanteile einer Signalform mehr Gewichtung erlangen. Anschließend wird die Sequenz mittels des DTW-Musterabgleichs mit Mustern verschiedener Programmabläufe verglichen. Die Muster, hier auch als Modelle bezeichnet, werden aus den vorherigen Signalverläufen gewonnen. In der Implementierung wird hierfür die erste (vom Nutzer annotierte) Signalform eines Programmablaufs verwendet. Zukünftig könnte ein Kriterium aufgestellt werden, ab dem die Erkennung zurückgewiesen wird und die Nutzerin bzw. der Nutzer zur Eingabe eines Programmnamens aufgefordert wird. Auf diese Weise könnte auf eine Initialisierungsphase verzichtet werden. Die oben genannte Logarithmierung muss auch auf die Muster angewendet werden. Ausgewählt wird der Programmablauf, der die geringste Abweichung zwischen der betrachteten Sequenz und dem entsprechenden Muster zeigt.

| Bedingung | Zustand „Aus“<br>max | Zustand „Ein“<br>min |
|-----------|----------------------|----------------------|
|           | 2 Watt               | 12 Watt              |
|           | $D_j = 5 \text{ S}$  | $D_j = 75 \text{ S}$ |

Abbildung 5.7: Parameter des Schwellenwert-Klassifikators, der dem DTW vorangestellt ist. S steht hier wieder für eine Samplelänge.

Die oben benannten Sequenzen wurden mittels eines vorgeschalteten Schwellenwert-Klassifikators (siehe Unterabschnitt 5.2.6) aus einem Energieverbrauchssignal separiert. Eine Sequenz stellt jeweils alle Messdaten eines produktiven Zustands dar. Die Messdaten der nicht-produktiven Zustände wurden verworfen. Wenn eine geeignete Parametrisierung des Schwellenwert-Klassifikators gewählt wurde, beinhalten die Sequenzen jeweils genau einen in sich geschlossenen Programmablauf des Gerätes. In Zusammenhang mit dem DTW wurde dazu die Parametrisierung aus Abbildung 5.7 verwendet. Bei dieser verwendeten Parametrisierung müssen zwei Gerätenutzungen bei sekundlicher Abtastung mindestens 75 Sekunden auseinanderliegen. Andernfalls werden die Aktivitäten nicht korrekt voneinander separiert.

### 5.2.9 Mehrgeräteerkennung mittels künstlicher neuronaler Netze anhand von Netzspannungsbeeinflussungen

In Unterabschnitt 4.2.5 wird eine Zustandsidentifikation mehrerer Geräte vorgeschlagen. Hierzu wurden in Voruntersuchungen verschiedene Datenquellen betrachtet, die die geforderten Orthogonalitäten aufweisen. Dabei wurde festgestellt, dass kleinste Netzspannungsbeeinflussungen im Sinne der Zustandsidentifikation eine betrachtenswerte Messgröße darstellen (siehe u. a. Bachelorarbeit [14/704]). Viele elektrische Geräte emittieren in ihrem Betrieb elektromagnetische Störungen (engl. electromagnetic interference, EMI) zurück in das Energieversorgungsnetz. Sie prägen sich als Beeinflussungen der lokalen 230-V-Versorgungsspannung aus, im Folgenden auch als Netzspannungsbeeinflussungen bezeichnet. Diese Beeinflussungen sind in der Regel unerwünscht, da sie andere Geräte und Systeme in ihrem Betrieb beeinträchtigen können. Die Stärke der Beeinflussungen hängt vom jeweiligen Gerät und der Güte der Netzimpedanz ab. In diesem Unterabschnitt sollen diese emittierten Signale für eine Zustandsidentifikation gleich mehrerer aussendender Geräte genutzt werden. Diese Idee ist nicht neu. Sie wurde beispielsweise in [Fro+11, S. 35f] vorgeschlagen.

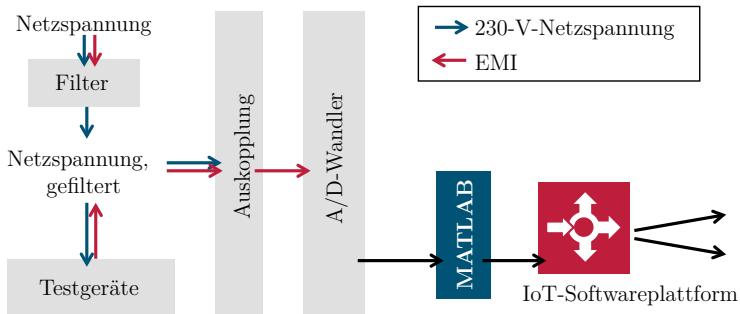


Abbildung 5.8: Mess- und Demonstrationsaufbau zur Aufnahme von Netzspannungsbeeinflussungen (EMI). Mit der Auskopplung werden die auszuwertenden Signalanteile aus der Netzspannung gewonnen. Für die Erfassung der Messsignale der einzelnen Geräte entfernt der eingangsseitige Filter die Beeinflussungen anderer Geräte aus der Netzspannung, sodass in diesem Bereich nur noch die Beeinflussungen der lokalen Geräte vorliegen.

In den Voruntersuchungen wurde zudem deutlich, dass insbesondere die hohen Frequenzanteile interessant für eine Erkennung sind. Entsprechend wird nachfolgend der Frequenzbereich von 100 MHz bis 1 GHz fokussiert. Für einen Mess- und Demonstrationsaufbau wurde eine AD-Wandler-Karte mit einer Auflösung von 16 Bit verwendet. Mit dieser können Frequenzen bis 1 GHz aufgenommen werden. Ein Adapter zur Auskopplung der elektromagnetischen Beeinflussungen aus der 230-V-Netzspannung verbindet den AD-Wandler und die Netz-

spannung. Der Adapter besteht aus einem Hochpass-Filter, der das Messsignal vom 230-V-Netzspannungspotential abtrennt. Dieser Aufbau ist in Abbildung 5.8 dargestellt. Dem Messaufbau ist ein Tiefpassfilter vorgeschaltet. Dadurch können die Beeinflussungen einzelner (oder auch mehrerer) Geräte ohne die Beeinflussungen des restlichen Niederspannungsnetzes aufgenommen werden.

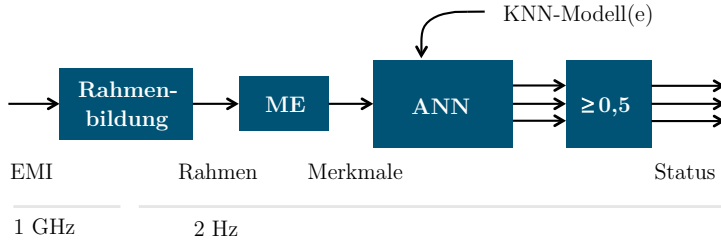


Abbildung 5.9: Struktur der Erkennung von mehreren Geräten anhand von künstlichen neuronalen Netzen (ANN). Die Struktur besteht zusätzlich aus einer Rahmenbildung und einer Merkmalsextraktion. Für die Merkmalsextraktion wird eine Fast-Fourier-Transformation verwendet.

Für die Zustandsidentifikation wird auf Basis der in Unterabschnitt 4.2.5 beschriebenen Struktur der Aufbau aus Abbildung 5.9 verwendet. Für die Merkmalsextraktion wurde eine *schnelle Fourier-Transformation* (engl. Fast Fourier Transformation, FFT) gewählt, damit die einzelnen Frequenzanteile der Netzspannungsbeeinflussungen als getrennte Merkmale auftreten. Der Merkmalsextraktion ist eine Rahmenbildung vorgeschaltet, wodurch Daten zusammengefasst und nachfolgend gemeinsam verarbeitet werden. Diese ist für die FFT notwendig. Die Erkennung greift auf künstliche neuronale Netze zurück, wie in Unterabschnitt 4.2.4 beschrieben. Die Eingangsneuronen entsprechen den berechneten Merkmalen aus der FFT-Analyse. Für jedes zu erkennende Gerät wurde ein Ausgangsneuron verwendet. Für die Gerätezustände wurden nur zwei Betriebszustände unterschieden: Der Zustand „Aus“ wurde dem Wert Null und der Zustand „Ein“ dem Wert Eins zugeordnet. Da die Ausgangsneuronen kontinuierliche Werte liefern, wurde jedem Ausgangsneuron eine Entscheidungsschwelle mit dem Wert 0,5 zugeordnet. Die verwendeten künstlichen neuronalen Netze wurden aus Modellen von verschiedenen Geräten trainiert, die aus getrennten Messungen gewonnen wurden. Für das Training werden die notwendigen Kombinationen durch Addition der Merkmale gewonnen. Dadurch wird die in Unterabschnitt 4.2.5 vorgeschlagene Reduktion der Trainingsdaten realisiert.

Eine Wiederholrate dieser Analyse wurde nicht festgelegt. Eine Festlegung ist nicht notwendig, da (bisher) keine der nachfolgenden Verarbeitungselemente vorherige Messdaten oder Ergebnisse in ihre Berechnungen einbezieht. Für den Live-Betrieb wurde die Wiederholrate so

gewählt, dass die Zustandsidentifikation auf dem verwendeten Rechner ohne Unterbrechung erfolgte. Dabei wurde eine Wiederholrate von ungefähr 2 Hz realisiert.

Die beschriebene Verarbeitungskette wurde in der Simulationsumgebung MATLAB und ein Teil der Hilfsprogramme zur Datenaufnahme des AD-Wandlers in der Programmiersprache C realisiert. Im Live-Betrieb werden die erkannten Zustände zusätzlich über die IoT-Softwareplattform veröffentlicht.

Der Live-Demonstrator wurde gemeinsam mit einem Studenten in der Masterarbeit [16/015] entwickelt.

## 5.3 Exemplarische Anwendungen zum Energiemanagement

In diesem Abschnitt wird die Implementierung einer Auswahl von Energiemanagement-Anwendungen dargestellt, die auf Basis der entwickelten Plattform realisiert wurden. Diese Energiemanagement-Anwendungen besitzen generell einen exemplarischen Charakter. Sie wurden entwickelt, um zu zeigen, ob die entwickelte Plattform in der Lage ist, die gewünschten Anwendungen auszuführen. Die Implementierungen haben ihre Verwendbarkeit in der Praxis bewiesen; sie erheben aber nicht den Anspruch, jeden Betriebsfall abzudecken.

Die Anwendungen wurden auf Basis der Funktionalitäten aus Abschnitt 4.4 bis 4.6 gegliedert. In Unterabschnitt 5.3.1 werden zunächst Nutzer-unterstützende Funktionalitäten beleuchtet. Anschließend werden ab Unterabschnitt 5.3.2 die Umsetzungen der Funktionalitäten zum automatisierten Energiemanagement beschrieben. Dabei wird zunächst auf die Funktionalität der direkten Verknüpfung eingegangen. In Unterabschnitt 5.3.3 wird die Umsetzung der Funktionalität zur Steuerung auf Basis von Schwellenwerten erläutert. Die Funktionalität zur Abstimmung von Geräten untereinander im Sinne der Eigenverbrauchsoptimierung wird in Unterabschnitt 5.3.4 umgesetzt. In Unterabschnitt 5.3.5 werden die Funktionalitäten jenseits des Energiemanagements realisiert.

### 5.3.1 Nutzer-unterstützende Funktionalitäten zum Einsparen von Energie

Aufbauend auf den Überlegungen in Abschnitt 4.4 wurden verschiedene Anwendungen für Smartphones, Tablets oder Großbildschirme implementiert. Die Anwendungen wurden in HTML5 umgesetzt und sind damit auf allen genannten Gerätearten sowie auch auf Desktop-



Rechnern einsetzbar. Dennoch wurden einige der später vorgestellten Anwendungen nur für spezifische Auflösungen optimiert.

Für die Umsetzung wurde die Erweiterung der IoT-Softwareplattform zur Darstellung von Nutzeroberflächen verwendet (siehe Unterabschnitt 5.1.6). Die einzelnen Anwendungen wurden jeweils als separate Software-Anwendungen der IoT-Softwareplattform implementiert. Die Anwendungen enthalten selbst keine anzuzeigenden Daten, sondern erhalten diese über die IoT-Softwareplattform. Um eine aussagekräftige Anzeige zu erhalten, müssen die Anwendungen also in einer Umgebung mit eingebundenen Geräten gestartet werden. Um besondere Funktionen oder Analysen in den Anwendungen zu ermöglichen, sind bedarfsweise weitere spezifische Software-Anwendungen in die Umgebungen einzubinden.

| Name              | Meta              |
|-------------------|-------------------|
| /Küche            | Bereich/Raum      |
| /Energieverbrauch | Daten/Energie     |
| /Spülmaschine     | Bereich/Gerät     |
| /Energieverbrauch | Daten/Energie     |
| /Status           | Status/Gerät      |
| /Wasserkocher     | Bereich/Gerät     |
| /Energieverbrauch | Daten/Energie     |
| /Status           | Status/Gerät      |
| /Kochzeit         | Daten/Statuslänge |
| /Wohnzimmer       | Bereich/Raum      |
| /Energieverbrauch | Daten/Energie     |
| /Schreibtisch     | Bereich/Tisch     |
| /Energieverbrauch | Daten/Energie     |
| /Drucker          | Bereich/Gerät     |
| /Energieverbrauch | Daten/Energie     |
| /Status           | Status/Gerät      |

Abbildung 5.10: Exemplarische Struktur an Datenressourcen. Repräsentiert werden zwei Räume mit insgesamt drei Geräten. Der Schreibtisch ist ein Unterbereich des Wohnzimmers. Räume und Geräte besitzen Datenressourcen für den Energieverbrauch. Die Geräte (grau hinterlegt) besitzen zusätzlich jeweils eine Datenressource, die den Status des Geräts beinhaltet. Für den Wasserkocher wurde zusätzlich eine weitere Analyse aktiviert, mit der die letzten Kochzeiten visualisiert werden können. Die Spalte Meta zeigt den ins Deutsche übersetzten Typ der Datenressource an, der in den Metadaten hinterlegt ist. Es werden die Grundtypen „Bereich“ für Navigationselemente, „Status“ für Datenobjekte und „Daten“ für Datenreihen unterschieden. Letztere sollen als Zeitreihen visualisiert werden können.

Die Anwendungen benötigen keine Konfiguration. Sie zeigen die Datenressourcen auf Basis ihrer hierarchischen Struktur und ihrer Metadaten an. Für eine sinnvolle Untergliederung in mehrere Ebenen bzw. Unterbereiche sollten die Datenressourcen einer entsprechenden

hierarchischen Baumstruktur folgen. Abbildung 5.10 zeigt exemplarisch eine solche Baumstruktur. Die Spalte Meta zeigt die übersetzten Typen der Datenressourcen an, anhand derer die Datenressourcen visualisiert werden. Deutlich wird, dass alle Datenressourcen des Typs „Bereich“ als Navigationselemente verwendet werden können. Jeder Bereich besitzt zudem eine Datenressource zum Energieverbrauch des „Bereichs“.

### Anwendung: Das Energiecockpit

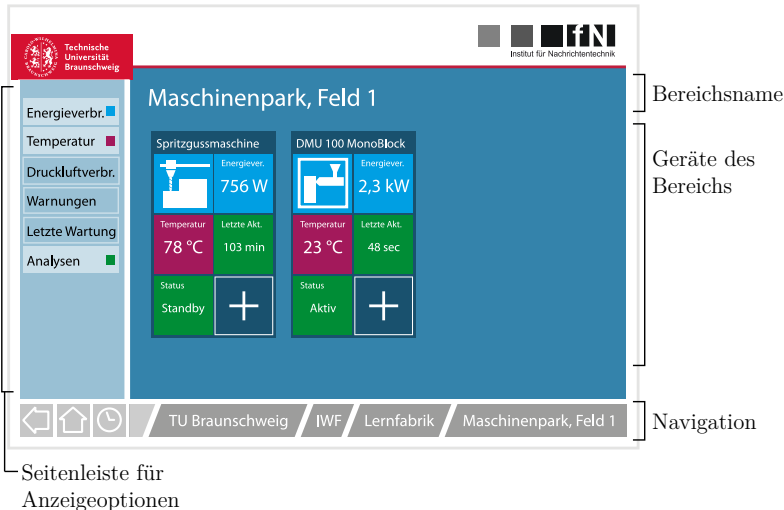


Abbildung 5.11: Die Display-Anwendung „Energiecockpit“ gibt einen Überblick über Geräte und ihre Energieverbräuche in einer Umgebung. Dargestellt werden zwei Maschinen mit Informationen zum Energieverbrauch, zu Temperaturen, zum aktuellen Betriebsstatus sowie zur Zeitdauer des letzten produktiven Betriebsstatus.

Das „Energiecockpit“ soll im Sinne der Überlegungen aus Unterabschnitt 4.4.1 einen Überblick über eine größere Umgebung ermöglichen. Es ist Teil der Anwendung energycockpit. Die Hauptansicht ist in Abbildung 5.11 dargestellt. Sie besteht aus virtuellen Kacheln, die die Unterobjekte und ihre Eigenschaften anzeigen. In der Abbildung handelt es sich bei den Unterobjekten um zwei Maschinen. Mit einem Klick auf den Namen eines Objekts oder auf das Symbolbild kann die Visualisierungsebene gewechselt werden. Handelt es sich bei dem Objekt um eine Datenressource vom Typ „Bereich“, wird dieser einschließlich der Unterobjekte in der Hauptansicht visualisiert. Handelt es sich bei dem Objekt um eine Datenreihe, wird diese als Graphansicht dargestellt. Über die Metadaten können zusätzlich Einheiten und Skalierungen für die Graphansicht vorgegeben werden. Weiterhin gibt es eine große Auswahl spezieller Ansichten mit der weitere Daten im Energiecockpit dargestellt

werden können, darunter z. B. die Visualisierung von Bedienelementen, Bilddateien und weiteren spezifischen Datenobjekten.

Eine Bedienerin oder ein Bediener der Anwendung kann zudem über die Anwendung auch weitere Analysen zu jedem Objekt hinzufügen und damit weitere Visualisierungen in der Anzeige aktivieren. Diese Analysen werden als Anwendungen auf dem lokalen Koordinator der Datenressource des ausgewählten Objekts gestartet. Voraussetzung ist, dass auf diesem lokalen Koordinator zuvor die notwendigen Rechte eingeräumt wurden. Als Analyse wurde beispielsweise die Anwendung **analysis/histogram** implementiert, die die Berechnung und Visualisierung von Histogrammen aus Datenreihen ermöglicht.

Das „Energiecockpit“ wurde ursprünglich als Display-Anwendung für einen Großbildschirm mit Touchbedienung der Lernfabrik des Instituts für Werkzeugmaschinen und Fertigungstechnik (IWF) entworfen. Damit war das Ziel verbunden im Sinne einer Leitwarte vor Ort schnell eine Übersicht über den Energieverbrauch der Werkzeugmaschinen zu ermöglichen. Die Anwendung lässt sich auch als Tablet- oder Desktop-Anwendung verwenden. Neben dem in Abbildung 5.11 dargestellten Design, das dem Corporate Design der TU Braunschweig folgt, steht auch ein markenloses Design zur Verfügung.

Die Anwendung wurde in der Masterarbeit [16/011] entwickelt und mit einer Umfrage bei Anwenderinnen und Anwender evaluiert.

#### **Anwendung: Meine Energie (unmittelbare Rückmeldungen)**

Die Anwendung **me** für „Meine Energie“ basiert auf den Überlegungen aus Unterabschnitt 4.4.2. Sie verfolgt das Ziel, Anwenderinnen und Anwendern einer lokalen (hier privaten) Umgebung eine Rückmeldung über ihre Gerätenutzung und die korrespondierenden Kosten zu geben.

Eine Anwenderin oder ein Anwender kann über die in Abbildung 5.12 dargestellte Navigationsstruktur Geräte auswählen und die letzten Gerätenutzungen visualisieren. Dabei werden neben den Zeiten und der Länge der jeweiligen Nutzung auch die Kosten dargestellt. Zusätzlich werden auch laufende Gerätenutzungen angezeigt. Weiterhin kann die Anwenderin oder der Anwender über eine Abonnement-Funktion unmittelbar über neue abgeschlossene Gerätenutzungen informiert werden.

Die Anwendung wurde in der Masterarbeit [16/003] entworfen und mit einer Umfrage bei Anwenderinnen und Anwendern evaluiert. Weitere Nutzer-unterstützende Anwendungen wurden in den Masterarbeiten [13/007], [14/003] und [15/002] sowie in dem Medientechnischen Projekt [13/703] umgesetzt und evaluiert.

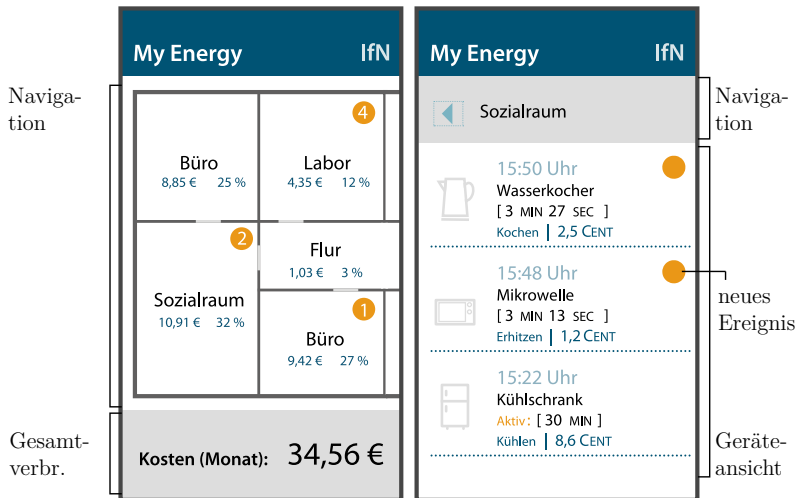


Abbildung 5.12: Die Smartphone-Anwendung „Meine Energie“ gibt der Nutzerin bzw. dem Nutzer direkte Rückmeldungen über ihre bzw. seine Gerätenutzung. Die linke Ansicht zeigt die Navigationsstruktur in Form einer Karte. Mittels Berührung eines Bereichs wird dieser dargestellt (rechte Ansicht). Dabei werden die Geräte in der Reihenfolge ihrer letzten Nutzung angezeigt. Neben der Nutzungszeit und -länge werden die jeweiligen Kosten der einzelnen Nutzungen angezeigt. Weiterhin können durch Auswahl eines einzelnen Gerätes die davorliegenden Nutzungen angezeigt werden.

### 5.3.2 Automatisiertes Energiemanagement mittels direkter Verknüpfung

Beim automatisierten Energiemanagement nach Abschnitt 4.5 sollen Geräte optimiert durch eine Software-Anwendung gesteuert werden. Die dazu notwendigen Steuerungsbefehle wurden über die RPC-Funktion `set(state)` mit der Übergabe eines neuen gewünschten Zustandes im Rahmen der Geräteeinbindung (vgl. Unterabschnitt 5.1.8) realisiert. Damit sind grundsätzlich alle Software-Anwendungen mit passenden Rechten in der Lage die Geräte anzusteuern.

Mit der Funktionalität der direkten Verknüpfung nach Unterabschnitt 4.5.1 wird die Möglichkeit eröffnet, ein Gerät direkt aufgrund eines Ereignisses oder einer Aufforderung zu steuern. Die nachfolgenden Hilfsanwendungen stellen diese Funktionalität bereit. Die Hilfsanwendungen können dazu direkt oder auch als Basis für die Vererbung verwendet werden.

- Anwendung **rpc-link**: Verknüpfung einer Datenressource als Ereignisquelle mit einer zu steuernden Datenressource. Die Anwendung abonniert dazu die Datenressource der

Ereignisquelle und löst den Steuerungsbefehl in Form der RPC-Funktion **set** auf der zu steuernden Datenressource aus.

- Anwendung **forward**: Bidirektionale Verknüpfung von zwei Datenressourcen. Dabei wird zusätzlich auch die zu steuernde Datenressource abonniert und ihre Ereignisse als Steuerungsbefehl an die erste Datenressource gesendet. Eine solche Verknüpfung wird auch als Zwei-Wege-Verknüpfung bezeichnet. Diese ist z. B. notwendig, wenn ein Schalter mit Statusanzeige den Status eines zu schaltenden Gerätes anzeigen soll. So wechselt die Statusanzeige auch dann ihre Anzeige, wenn das Gerät direkt durch die Nutzerin oder den Nutzer bedient wird. Angewendet auf zwei Geräte schalten beide in den gleichen Zustand, unabhängig davon, welches Gerät bedient wird.

Um Maßnahmen, wie die Steuerung von Geräten aufgrund eines Direkteingriffs oder des Lastmanagements zu realisieren (siehe Unterabschnitt 2.1.4), werden weitere zusätzliche Funktionen benötigt. Deshalb wurden vereinfachte Beispielanwendungen dieser Maßnahmen umgesetzt, die nachfolgend beschrieben werden.

#### **Anwendung: Abstimmung von Geräten untereinander**

Durch die Software-Anwendung **event-device** werden ein oder mehrere Geräte in Abhängigkeit des Gerätestatus eines anderen Gerätes gesteuert. Dies realisiert die Maßnahme zur Abstimmung von Geräten untereinander (siehe Unterabschnitt 2.1.4). Üblicherweise werden hierbei Geräte ein- bzw. ausgeschaltet, wenn andere Geräte ihren Dienst benötigen bzw. nicht mehr benötigen.

Die Anwendung erbt die Funktionen der Anwendung **rpc-link**. In Erweiterung zu dieser Anwendung übersetzt die Anwendung **event-device** verschiedene Datenstrukturen zum Betriebszustand. Zusätzlich kann auch eine Übersetzung der Zustände vorgenommen werden, wodurch z. B. selektiv nur auf das Ein- oder das Ausschalten reagiert werden kann.

Zur Realisierung dieser Funktionalität muss die Anwendung manuell von der Administratorin oder vom Administrator konfiguriert werden. Dazu muss eine Ereignisquelle in Form einer Datenressource gewählt werden, die den Status eines Geräts abbildet. Zusätzlich muss eine Datenressource eines zu steuernden Geräts gewählt werden. Dabei kann auch eine Übersetzung der Zustände definiert werden. Die Software-Anwendung kann dazu in jedem lokalen Koordinator initialisiert werden, der Zugriff auf beide Datenressourcen besitzt.

#### **Anwendung: Direkteingriff im Sinne des Market Demand Response**

Beim Direkteingriff im Sinne des Market Demand Response führt die Energievertriebsgesellschaft, mit der ein Kunde bzw. eine Kundin einen Vertrag über den Bezug von elektrischer Energie abgeschlossen hat, Eingriffe durch, die spezifische Geräte der Kundin bzw. des

Kunden steuern (siehe Unterabschnitt 2.1.2). Im Fall der Sondervertragskunden wird die schaltbare Leistung eventuell summiert über alle Übergabestellen vertraglich festgehalten. Diese Eingriffsmöglichkeiten werden dem Kunden in der Regel finanziell honoriert.

Im Gegensatz zur beschriebenen vertraglichen Festlegung von Geräten, wird in dem hier dargestellten Demonstrator angenommen, dass nicht jedes Gerät zu jeder Zeit für einen Eingriff zur Verfügung steht. Die Vertriebsgesellschaft soll jedoch weiterhin in der Lage sein, spezifische Geräte (oder Gerätegruppen) zu steuern. Zusätzlich soll das System der Vertriebsgesellschaft hier dynamisch melden, welche Geräte wann, wie lange und mit welcher Leistung beeinflusst werden können.

Wie in Unterabschnitt 4.5.1 beschrieben, soll die dynamische Meldung von Geräten durch die Initialisierung der Datenressource über die hierarchische Kommunikationsstruktur realisiert werden. Die Initialisierung beinhaltet dabei die Metadaten-Einträge `dr_power` für die abschaltbare Leistung in Watt und `dr_duration` für die Länge der möglichen Beeinflussung in Sekunden. Bei einer zuschaltbaren Leistung muss der Wert für die Leistung mit negativen Vorzeichen angegeben werden. Damit die Metadaten über die hierarchische Kommunikationsstruktur zur Vertriebsgesellschaft übermittelt werden können, muss einer der übergeordneten lokalen Koordinatoren eine Kommunikationsverbindung zu einem lokalen Koordinator der Vertriebsgesellschaft besitzen. Weiterhin müssen alle zwischengeschalteten lokalen Koordinatoren die Weiterleitung der Datenressource und ihrer Metadaten hin zur Vertriebsgesellschaft erlauben. Die Vertriebsgesellschaft kann dann über die RPC-Funktion `set` einen Eingriff auslösen. Dazu muss in allen zwischengeschalteten lokalen Koordinatoren auch der Aufruf dieser RPC-Funktion auf der Datenressource erlaubt sein.

Der Demonstrator besteht dazu aus den folgenden Komponenten:

- Mit der Anwendung `market-dr/device` kann diese Initialisierung umgesetzt werden. Bei der Konfiguration muss die Administratorin oder der Administrator sowohl die Datenressource des zu steuernden Gerätes sowie die Werte für die beiden Metadaten-Einträge angeben. Zukünftig könnten die Werte auch aus der Datenressource des Gerätes entnommen werden, sofern sie der Geräteeinbindung dieses Geräts bekannt sind. Die Anwendung protokolliert zusätzlich die erfolgten Eingriffe zur Ansicht durch die Eigentümerin oder den Eigentümer.
- Die Anwendung `market-dr/master` soll exemplarisch die Anwendung zur Ausführung bei der Vertriebsgesellschaft aufzeigen. Sie erhält von einer anderen Instanz oder von einer Mitarbeiterin oder einem Mitarbeiter der Vertriebsgesellschaft über die RPC-Funktion `dr(power, duration)` die Anweisung den Energieverbrauch um einen bestimmten Leistungsbetrag (dt:power in Watt) für eine bestimmte Zeit (dt:duration in Sekunden)

zu verringern oder zu erhöhen. Über die Funktion `subscribe_announcements` werden alle Datenressourcen der Anwendung `device-market-dr/device` identifiziert. Ein Administrator oder eine Administratorin kann zusätzlich Prioritäten für die Verwendung einzelner Geräte vorgegeben. Über die RPC-Funktion `set` werden Eingriffe ausgelöst, bis ein bestimmter Leistungsbetrag erreicht wird. Nicht mehr benötigte vorher getätigte Eingriffe werden beendet.

- Um die Privatsphäre bzw. Betriebsgeheimnisse besser zu schützen, können mit Hilfe der Anwendung `market-dr/aggregate` Geräte zu größeren Regeleinheiten zusammengefügt werden. Dabei ist zu beachten, dass sich die Eingriffsdauern nicht zu stark voneinander unterscheiden. Dies kann durch mehrere nicht überlappende Aggregationen auf einer Ebene ermöglicht werden. Im Unterschied zu den Geräten wird die Aggregation ebenfalls mit der RPC-Funktion `dr` und dem gewünschten Leistungsbetrag und der Zeit angesteuert.

Dieser Demonstrator wählt Geräte anhand der Prioritäten sowie einer genügend langen Eingriffszeit aus. Eine Optimierung der Geräteauswahl darüber hinaus findet nicht statt.

### 5.3.3 Steuerung auf Basis von Schwellenwerten

Wie in der Konzeption in Unterabschnitt 4.5.2 beschrieben, soll auch aufgrund von Messdaten und anderen nicht binären Datenquellen eine Steuerung realisiert werden.

Mit der Hilfsanwendung `rpc-link-threshold` lässt sich diese Funktionalität auf Basis von Schwellenwerten ermöglichen. Sie wurde abgeleitet von der zuvor benannten Anwendung `rpc-link`, besitzt jedoch zusätzlich eine Zuordnung anhand von einstellbaren Schwellenwerten.

Nachfolgend werden wiederum verschiedene Beispielsanwendungen auf Grundlage der in Unterabschnitt 2.1.4 beschriebenen Maßnahmen vorgestellt.

#### Anwendung: Lastspitzenreduktion

Für die Demonstration der Lastspitzenreduktion wurde die Anwendung `peak-manager` implementiert. Diese stellt den Quotienten aus aktueller prognostizierter Viertelstundenleistung und dem gewünschten maximalen Grenzwert als Datenressource zur Verfügung (siehe auch Unterabschnitt 4.5.2). Auf diesen Wert, der sich in der Regel im Bereich zwischen den Werten 0 und 1 bewegt, kann individuell pro Gerät mit Hilfe der Anwendung `peak-manager/device` reagiert werden. In der Anwendung kann der Schwellenwert `limit` festgelegt werden, ab dem das Gerät abgeschaltet wird. Im Unterschied zur Hilfsanwendung

**rpc-link-threshold** sucht diese Anwendung die benötigte Datenressource des Quotienten mit Hilfe der Funktion `subscribe_announcements` selbst.

Für die Demonstration wird anstelle der tatsächlichen Messung in der Zuleitung zum Netzbetreiber hier die Aufsummierung aller aktuellen Leistungen der verbundenen Geräte mit Hilfe der Anwendung **aggregation** verwendet. Um für eine Lastspitzenreduktion auch nicht eingebundene Geräte zu beachten, sollte in der Praxis jedoch auf eine reale Messung zurückgegriffen werden.

### Anwendung: Variable Tarife

Die in Unterabschnitt 2.1.4 benannten variablen Tarife sind eine vielbenannte Maßnahme, wenn es um die in Beeinflussung von Geräten in Privathaushalten geht (vgl. [AE07, S. 2 und S. 5]). Der im Folgenden vorgestellte Demonstrator zur Nutzung von variablen Tarifen besteht aus zwei Anwendungstypen:

- In einem lokalen Koordinator der Vertriebsgesellschaft werden Echtzeitpreise in die hierarchische Kommunikationsstruktur eingebracht. In diesem Demonstrator stehen dazu die Anwendungen **em-real-time-pricing/price-random** für zufallsverteilte Echtzeitpreise und **em-real-time-pricing/price-schedule** für eine wiederkehrende Abfolge von Energiepreisen zur Verfügung. Über die Option `interval` kann die Wiederholrate der Veröffentlichung der Preise eingestellt werden. Für die erstgenannte Anwendung kann zusätzlich das Intervall der Echtzeitpreise und für die zweitgenannte Anwendung können die Werte der Abfolge der Echtzeitpreise angegeben werden.
- Die Anwendung **em-real-time-pricing/event-price** kann verwendet werden, um ein Gerät oder eine andere Datenressource ein- bzw. auszuschalten. Dazu müssen obere und untere Preisgrenzen für die Zu- bzw. Abschaltung hinterlegt werden. Unter der Angabe von Preisgrenzen mit negativem Vorzeichen kann die Anwendung auch auf die Ab- bzw. Zuschaltung von Energieerzeugern angewendet werden.

## 5.3.4 Abstimmung von Geräten untereinander anhand von Optimierungsalgorithmen

Bei Einsatz von Energieerzeugern ist die Eigenverbrauchsoptimierung in der Regel besser geeignet als die zuvor dargelegten Anwendungen, da Geräte auch untereinander abgestimmt werden können. Wie in Unterabschnitt 4.5.3 vorgeschlagen, soll dazu ein iterativer, verteilter Optimierungsansatz verwendet werden:



### Anwendung: Eigenverbrauchsoptimierung

Der Demonstrator zur Eigenverbrauchsoptimierung folgt dem in Unterabschnitt 4.5.3 beschriebenen Schema. Die Anwendung **em-prosumer/forecast** realisiert den beschriebenen Austausch von Energieverbrauchs-Prognosen. Dazu muss die Anwendung in jedem lokalen Koordinator, der an der Eigenverbrauchsoptimierung teilnehmen soll, mindestens einmal gestartet werden.

Die summierte Vorhersage aller untergeordneten Systeme wird in der Datenressource der Anwendung veröffentlicht. Die Vorhersagen bestehen aus Leistungsangaben in Watt für jede Minute der nächsten 24 Stunden also aus 1.440 Werten. Die Datenressource kann von übergeordneten Instanzen der gleichen Anwendung abonniert werden. Mit Aufruf der RPC-Methode **top\_set\_forecast(forecast)** sendet eine übergeordnete Instanz die summierte Vorhersage aller anderen Systeme zurück an die untergeordneten Instanzen.

Die Optimierung eines oder mehrerer Geräte wird separat in der Anwendung **optimizer** durchgeführt. Für die Optimierung wird ein evolutionärer Algorithmus verwendet. Die Optimierung wird von der Anwendung **em-prosumer/forecast** durch den RPC-Aufruf **em\_set\_forecast(forecast)** angestoßen. Dabei wird die summierte Vorhersage aller fremden unter- und übergeordneten Instanzen übergeben. Die durch den Optimierungsalgorithmus berechnete Vorhersage wird als Rückgabewert der RPC-Funktion an die Anwendung **em-prosumer/forecast** zurückgegeben. Weiterhin kann der Optimierungsalgorithmus eine Energieverbrauchs-Vorhersage eines Geräts in Abhängigkeit der übergebenen Parameter erfragen. Dies erfolgt über den RPC-Aufruf **opt\_get\_forecast(parameter)** auf die Datenressource eines zu optimierenden Geräts. Die Optimierung variiert dabei die übergebenen Parameter bis eine optimale Lösung gefunden wurde. Sollen die optimierten Parameter vom Gerät übernommen werden, müssen diese abschließend mit dem RPC-Aufruf **opt\_set\_plan(parameter)** an das Gerät übergeben werden.

Für die Gerätesimulation wurden drei Anwendungen nach den in Unterabschnitt 4.1.5 beschriebenen Gerätearten implementiert. Für die ebendort benannten halbautomatischen und manuellen Geräte können mit den Anwendungen **em-prosumer/semiautomatic-device** und **em-prosumer/manual-device** Leistungswerte für die verschiedenen Betriebszustände vorgegeben werden. Im Unterschied zu manuellen Geräten, die nicht beeinflusst werden können, kann der Programmablauf bei halbautomatischen Geräten in bestimmten Phasen pausieren. Als automatisches Gerät wurde zusätzlich ein virtuelles Kühlaggregat in der Anwendung **virtual-fridge** implementiert, das seine Kühlphasen beliebig verschieben und unterbrechen kann. Die Leistungsaufnahme, der Isolationsgrad, die Kühleffizienz, die thermische Masse und weitere Werte können dabei eingestellt werden. Die Anwendungen berechnen dabei Prognosen nach Unterabschnitt 4.1.6.

Dieser Demonstrator wurde gemeinsam mit einem Studenten in der Masterarbeit [17/017] entwickelt.

### 5.3.5 Funktionalitäten jenseits des Energiemanagements

Neben dem Energiemanagement ergeben sich viele Funktionalitäten, die ebenfalls auf Basis der IoT-Softwareplattform sinnvoll umgesetzt werden könnten. Im Folgenden werden vier Anwendungen aus den Funktionalitäten vorgestellt, die in Abschnitt 4.6 beschrieben bzw. skizziert sind.

#### Anwendung: Home Control mit Szenensteuerung

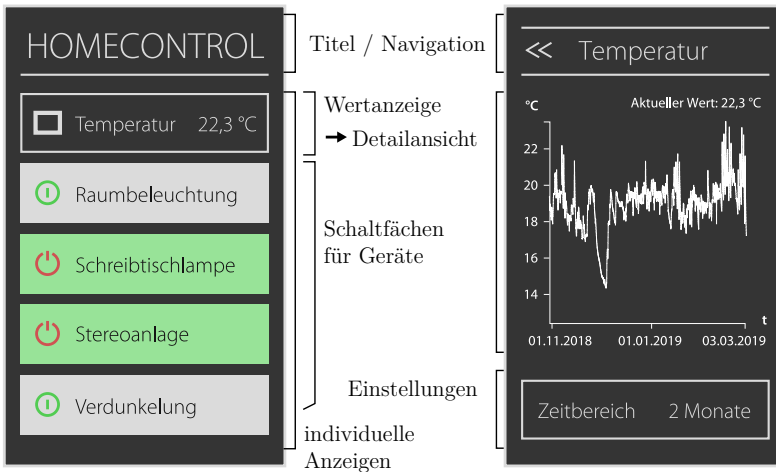


Abbildung 5.13: Darstellung der Smartphone-Anwendung Home Control. Die linke Ansicht zeigt die Navigationsansicht. Geräte können hier direkt über Schaltflächen geschaltet werden. Die rechte Ansicht zeigt eine Detailseite, die z. B. durch Klick auf die Wertanzeige „Temperatur“ erreicht werden kann.

In Analogie zu den Nutzer-unterstützenden Energiemanagementanwendungen wurde eine Nutzeroberfläche entwickelt, die auf die Steuerung von Geräten fokussiert. Diese ist in Abbildung 5.13 dargestellt. Sie ist Teil der Anwendung er-app-homecontrol. Über die Schaltflächen können Geräte direkt gesteuert werden. Auch Szenen (in der Abbildung nicht dargestellt) können ausgewählt werden. Die Ansteuerung erfolgt über die RPC-Funktion `set(state)`. Als Zielzustand `state` wird der invertierte Zustand der visualisierten Datenressource verwendet. Die rechte Ansicht zeigt eine Detailseite, die Datenreihen als Grafik visualisiert.

Für die Umsetzung der Szenen muss auf mindestens einem lokalen Koordinator die Anwendung scene bzw. scene-filter gestartet und konfiguriert werden. Bei der Konfiguration müssen verschiedene Szenennamen und die jeweils aktiven Geräte (sowie ggfs. deren gewünschte Zustände) hinterlegt werden.

### **Anwendungsskizze: Geräteüberwachung**

Für die Geräteüberwachung im Sinne der prädiktiven Instandhaltung können die Techniken aus Unterabschnitt 5.2.6 und folgende verwendet werden. Hier gibt es mehrere Anwendungsmöglichkeiten.

- Einerseits kann die Zustandsidentifikation genutzt werden, um (zusätzliche) fehlerhafte Zustände zu erkennen, die vom Regelbetrieb abweichen. Auf diese Ereignisse kann anschließend reagiert werden. Für die Zustandsidentifikation kann z. B. die Anwendung analysis/eventdetection genutzt werden.
- Als zweite Möglichkeit kann das Ergebnis einer Zustandsidentifikation mit dem tatsächlichen Gerätestatus eines vernetzten oder vollständig gesteuerten Gerätes verglichen werden. Auf Abweichungen kann dann reagiert werden.

Die Anwendungen rpc-link oder rpc-link-if können verwendet werden, um auf Ereignisse zu reagieren. So kann beispielsweise eine Warnlampe eingeschaltet oder mittels der Anwendung event-mail eine E-Mail versendet werden.

### **Anwendung: Automatische Abschaltung nach einer Zeitdauer**

Automatische Abschaltungen von thermischen Heizquellen, wie z. B. Herdplatten, Bügeleisen usw. können den Bewohnerinnen und Bewohnern eine zusätzliche Sicherheit geben. Dieser Beweggrund ist in Unterabschnitt 2.1.3 unter dem Begriff „ruhiges Gewissen“ beschrieben.

Die Anwendung device-timer ermöglicht eine solche automatische Abschaltung. Bei der Konfiguration muss der Anwendung eine Datenressource bekanntgegeben werden, die sich auf das abzuschaltende Gerät bezieht. Beim Start der Anwendung abonniert diese die Datenressource. Bei der Veröffentlichung von neuen Daten in der Datenressource werden alle bisher gestarteten Timer zurückgesetzt. Ist aus den Daten ersichtlich, dass das beobachtete Gerät eingeschaltet oder bedient wurde, wird ein neuer Timer gestartet. Nach Ablauf der konfigurierbaren Zeit wird mit dem RPC-Befehl `set` ein Befehl zum Ausschalten des Geräts an die Datenressource gesendet.

**Anwendung: Automatische Abschaltung bei Nutzerabwesenheit**

Für die Abschaltung bei Nutzerabwesenheit werden Informationen über die Anwesenheit einer Nutzerin oder eines Nutzers benötigt. Nachfolgend werden verschiedene Anwendungen vorgestellt, die auch miteinander kombiniert werden können.

- Die Anwendung **user-presence/switch** stellt einen virtuellen Schalter zur Verfügung, der von Nutzerinnen und Nutzern bedient werden kann. Der virtuelle Schalter kann auch mit realen Schaltern verknüpft werden. Über die RPC-Methode **set** können auch Verknüpfungen mit anderen Anwendungen hergestellt werden, wie z. B. die o.g. Anwendung **device-timer**.
- Die Anwendung **user-presence/ping** detektiert mit Hilfe regelmäßiger Ping-Anfragen, ob sich bestimmte Geräte in der Netzwerkumgebung befinden. Dabei kann sowohl auf erfolgreiche Ping-Antworten oder auf eine konfigurierbare Anzahl ausbleibender Ping-Antworten reagiert werden.
- Die Anwendung **user-presence-dhcpdiscover** reagiert auf DHCP-Anfragen, die Geräte aussenden, wenn sie dem Netzwerk betreten möchten. Dieser Fall tritt ein, wenn mobile Geräte in Reichweite des Funknetzwerk kommen oder feste Geräte eingeschaltet werden.

Alle drei Anwendungen stellen Datenressourcen zur Verfügung, die von anderen Anwendungen abonniert werden können. So können beispielsweise mit der Anwendung **rpc-link-if** ausgewählte Geräte automatisch ein- oder ausgeschaltet werden.

## 6 Einsatz und Evaluation des Systems in realen Umgebungen

In diesem Kapitel wird die Leistungsfähigkeit des Systemkonzeptes aus Kapitel 4 und dessen Umsetzung aus Kapitel 5 analysiert und anschließend beurteilt. Dabei soll zunächst festgestellt werden, ob die in Kapitel 3 dargestellten Defizite bezüglich der Einsatzfähigkeit bisheriger Systeme überwunden werden konnten. Weiterhin soll diskutiert werden, mit welchen Vor- und Nachteilen das gewählte Systemkonzept verbunden ist.

Im Rahmen dieser Arbeit wurde eine große Anzahl verschiedener Implementierungen erstellt (siehe Kapitel 5). Dies geschah aus zwei Gründen:

- Einerseits soll gezeigt werden, dass die konzipierte IoT-Softwareplattform als Grundlage verschiedenste Energiemanagement-Funktionalitäten und Geräteeinbindungsverfahren dienen kann. Aus diesem Grund werden in Abschnitt 6.1 zunächst einzelne Umgebungen und deren Randbedingungen vorgestellt, die im Rahmen dieser Arbeit mit Geräten oder Software ausgestattet wurden. Auch die realisierten Energiemanagementanwendungen werden vorgestellt. Damit soll bestätigt werden, dass die Einsatzfähigkeit in unterschiedlichen Umgebungen möglich ist.
- Andererseits soll beleuchtet werden, welche Konzepte und konkreten Umsetzungen für das Energiemanagement im Allgemeinen sinnvoll einsetzbar sind. In Abschnitt 6.2 bis 6.4 werden deshalb die Komponenten des Schichtenmodells (siehe Abschnitt 3.3) in der Reihenfolge mittlere, untere und obere Schicht auf Basis ausgewählter Analysen qualitativ und quantitativ untersucht. Dazu wird in Abschnitt 6.2 das Systemdesign und -verhalten betrachtet und beurteilt. In Abschnitt 6.3 werden die in Abschnitt 5.2 beschriebenen Umsetzungen zur Einbindung der Geräte anhand von realen Testdaten evaluiert. Die Zielfunktionalitäten zum Energiemanagement werden in Abschnitt 5.3 untersucht.

Abschließend wird in Abschnitt 6.5 ein Fazit aller Analysen mit Bezug auf das Systemkonzept präsentiert.

## 6.1 Analyse der Funktionsfähigkeit des Systemkonzepts anhand des Gesamtdemonstrators „Real-Life-Lab Campus 2020“

Im Rahmen des Projektes *EnEff Campus 2020*<sup>1</sup> wurde eine größere Anzahl von Einzeldemonstratoren zum Thema Energiemanagement auf Basis der IoT-Softwareplattform aufgebaut und schließlich zu einem Gesamtdemonstrator namens „*Real-Life-Lab Campus 2020*“ zusammengeschlossen. Damit wurde das Ziel verfolgt, das Verhalten von Teilsystemen sowie auch den Zusammenschluss von Teilsystemen in verschiedenen Umgebungen in der Praxis zu demonstrieren und zu untersuchen. Der Gesamtdemonstrator wurde in einem Workshop [6] des Projektes EnEff Campus 2020 vorgestellt.

Wie in Abschnitt 3.2 aufgezählt, unterliegen vorhandene Systeme für das Energiemanagement in der Regel Randbedingungen, die ihre Einsatzfähigkeit einschränken. Durch den Zusammenschluss realisierter Teilsysteme zum Gesamtdemonstrator sollte praktisch bestätigt werden, dass die IoT-Softwareplattform in unterschiedlichsten Umgebungen mit Hilfe verschiedener Umsetzungen aus Abschnitt 5.2 eingesetzt und verbunden werden kann, ohne dass Einschränkungen in den Umgebungen, den Einbindungen oder den Anwendungen auftraten. Auf Basis der IoT-Softwareplattform konnten so verschiedene Energiemanagement-Anwendungen in die Umgebungen eingebracht und untersucht werden.

Um die tatsächliche Einsatzfähigkeit zu verdeutlichen, wird in Unterabschnitt 6.1.1 zunächst die Struktur des Gesamtdemonstrators vorgestellt. Alle dort und folgend beschriebenen Elemente wurden auf Basis der in Abschnitt 5.1 beschriebenen IoT-Softwareplattform realisiert. In Unterabschnitt 6.1.2 werden die einzelnen Umgebungen näher charakterisiert. Dabei wird auch erläutert, welche Verfahren zur Einbindung von Geräten eingesetzt wurden. Die in den jeweiligen Umgebungen implementierten Energiemanagement-Anwendungen werden in Unterabschnitt 6.1.3 vorgestellt. Abschließend wird in Unterabschnitt 6.1.4 ein Fazit gegeben.

### 6.1.1 Gesamtaufbau und Kommunikationsstruktur

In Abbildung 6.1 ist die Struktur des Gesamtdemonstrators „Real-Life-Lab Campus 2020“ dargestellt. Die einzelnen Koordinatoren sind durch Systemgrenzen voneinander getrennt.

---

<sup>1</sup>Ein vom Bundesministerium für Wirtschaft und Energie (BMWi) gefördertes Projekt, welches das Ziel verfolgt, ein Maßnahmenpaket zur energetischen Optimierung des Campus für die deutsche Hochschul-landschaft zu entwickeln.

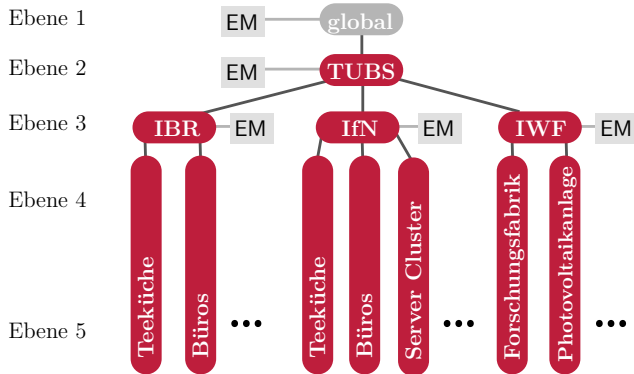


Abbildung 6.1: Für das Projekt EnEff Campus 2020 wurde ein Gesamtdemonstrator namens „Real-Life-Lab Campus 2020“ über alle Einzeldemonstratoren und erschlossenen Umgebungen realisiert. Das entstandene System überdeckt ausgewählte Umgebungen in verschiedenen Instituten der TU Braunschweig. Ebene 4 beinhaltet weitere Koordinatoren, die die Geräte in Ebene 5 zusammenfassen.

Um den Datenschutz zu gewährleisten, werden die Daten lokal in den Koordinatoren verarbeitet und nur notwendige Daten über diese Systemgrenzen weitergeleitet.

Alle Komponenten werden im Gesamtdemonstrator in fünf Ebenen abgebildet. Die zweitoberste Ebene besteht aus einem Koordinator für die gesamte TU Braunschweig. Dieser ist mit „TUBS“ bezeichnet. Darunter finden sich Ebenen mit Koordinatoren und Geräten, die dem TUBS-Koordinator entweder direkt oder über weitere Ebenen indirekt zugeordnet sind.

Direkt untergeordnet sind Koordinatoren für die angeschlossenen Institute: dem Institut für Betriebssysteme und Rechnerverbund (IBR), dem Institut für Nachrichtentechnik (IfN) und dem Institut für Werkzeugmaschinen und Fertigungstechnik (IWF). Der Koordinator für die TU Braunschweig hat die Aufgabe, die direkt untergeordneten Systeme zu koordinieren. Dazu sind auf diesem und den direkt untergeordneten Systemen Anwendungsteile zum Energiemanagement installiert, die in Abbildung 6.1 mit „EM“ bezeichnet sind.

Die einzelnen Institute sind wiederum mehrfach weiter unterteilt. Hier sind Koordinatoren für Gebäudeteile, Labore, Gerätearten wie Server und weitere zu finden. Einige dieser Komponenten werden exemplarisch im nächsten Unterabschnitt genauer dargestellt. Auch in den untersten Schichten befinden sich Anwendungsteile zum Energiemanagement, sofern dort Komponenten aktiv gesteuert werden sollen.

Zusätzlich existiert ein globaler Koordinator, der mit „global“ bezeichnet ist und dem Koordinator für die TU Braunschweig übergeordnet ist. Dieser Koordinator übernimmt

im Gesamtdemonstrator die Aufgabe, externe Vorgaben in das System einzubringen. Im Gesamtdemonstrator steht dieser Koordinator stellvertretend für die Anbindung von Systemen der Verteil- und Übertragungsnetzbetreiber und zu stadt- und länderübergreifenden administrativen Organen.

Die Ebenen vier und fünf besitzen keine speziellen Bedeutungen beispielsweise in Bezug auf den Datenschutz. Sie wurden lediglich so gewählt, um die Umgebungen und ihre Hierarchie abzubilden. Ersichtlich ist, dass in den tieferen Ebenen eine größere Anzahl paralleler Systeme ausgeprägt ist. Weiterhin kann angenommen werden, dass sich bei ähnlichem Aufbau der Organisationsbereiche auch die Aufgaben der einzelnen Koordinatoren einer Ebene ähneln.

Durch die hierarchische Struktur konnten erstmals Geräte in verschiedenen Umgebungen verteilt über den Campus der TU Braunschweig unter Einhaltung konkreter Anforderungen an den Datenschutz und die Datensicherheit miteinander verbunden werden.

### **6.1.2 Charakterisierung der Versuchsumgebungen und der sich ergebenden Herausforderungen**

Die erschlossenen Versuchsumgebungen weisen eine hohe Diversität in Bezug auf die eingebrachten Gerätearten auf, die den interdisziplinär aufgestellten Projektpartnern zu verdanken ist. Nachfolgend werden die Systeme der Institute beschrieben, die den jeweiligen Koordinatoren direkt oder indirekt zugeordnet sind:

Am Institut für Betriebssysteme und Rechnerverbund, einem Institut der Informatik, wurden

- die Geräte von vier Büros mit Energieverbrauchs-Messgeräten ausgestattet und mit den Verfahren aus Unterabschnitt 5.2.7 eingebunden. Der Energieverbrauch wurde jeweils im Intervall von zwei Sekunden ermittelt.
- Weiterhin wurde ein konventioneller Kühlschrank mit Hilfe eines Raspberry-Pi-Mini-Computers mit Energiemanagementfunktionen ausgestattet. Dazu wurden Sensoren für die Erfassung der Temperatur und die Öffnung der Tür sowie ein Schaltaktor für das Kühlaggregat verbaut. Mittels einer eigens implementierten Regelung kann der Kühlschrank nun selbstständig am Energiemanagement teilhaben, indem beispielsweise die Kühlphasen auf geeignete Zeiträume verschoben werden.<sup>2</sup>

Am Institut für Nachrichtentechnik, an dem diese Arbeit entstanden ist, wurden

---

<sup>2</sup>Näheres zu diesem Demonstrator kann der Bachelorarbeit [Hei16] entnommen werden.



- die Teeküche und zeitweise einige Büros mit Energieverbrauchs-Messgeräten ebenfalls mit den Verfahren aus Unterabschnitt 5.2.6 bzw. 5.2.7 eingebunden. Dabei wurden abweichend von denen im IBR verwendeten Geräten Produkte eines anderen Herstellers eingesetzt. Hier konnte der Energieverbrauch mit einem Zeitintervall von einer Sekunde bestimmt werden.
- Zusätzlich wurden einige Simulationsserver mit Energiemessfunktionen ausgestattet. Hier wurden Messgeräte und auch die in Unterabschnitt 5.2.4 beschriebenen Verfahren zur Schätzung des Energieverbrauchs eingesetzt.
- Mit einem weiteren Demonstrator wurde ein Laboraufbau aus elektronischen Geräten mit einem einzelnen hochabtastenden Spannungssensor vermessen. Anhand der elektromagnetischen Beeinflussungen, die die Geräte auf die Netzspannung aufrägen, konnte auf den Betriebsstatus der Geräte geschlossen werden. Dazu wurde die Implementierung aus Unterabschnitt 5.2.9 eingesetzt. Auch dieser Demonstrator wurde mit dem Gesamtdemonstrator verbunden.

Am Institut für Werkzeugmaschinen und Fertigungstechnik, einem Institut der Fakultät für Maschinenbau, wurden

- die Werkzeugmaschinen der dortigen Forschungsfabrik, einer Maschinenhalle mit großen aktiv genutzten Werkzeugmaschinen, mit verschiedenen Verfahren aus Abschnitt 5.2 eingebunden. Zusätzlich wurde auch die Gebäudeleittechnik (GLT), wie Beleuchtungs-, Kälte- und Druckluftsysteme angeschlossen. Auch die Energieerzeugung einer Solar- sowie einer Windkraftanlage werden erfasst. Die Messdaten wurden über Messtechnikmodule der Firma Beckhoff Automation GmbH & Co. KG erfasst und über die softwaremäßige Anbindung des EtherCAT-Feldbus-Systems bzw. des Modbus-Feldbuses angebunden (siehe Unterabschnitt 5.1.7). Die Messtechnikmodule lassen je nach Typ eine Abtastung des Energieverbrauchs von bis zu 50 Hz bzw. bei den Hochfrequenz-Modulen bis zu einem Kilohertz zu.
- Weiterhin wurden die Energieverbrauchsdaten der Experimentierfabrik des Instituts erschlossen. Hierbei handelt es sich um eine Fertigungsstraße im Miniaturformat. Teile dieses Demonstrators wurden unter Verwendung der IoT-Softwareplattform am Gemeinschaftsstand des Landes Niedersachsen auf der CeBIT 2016 ausgestellt.

Die eingebundenen Geräte bzw. Maschinen stellen nur eine kleine Auswahl der Geräte der Institute dar. Die Auswahl ist nicht repräsentativ, jedoch wird die große Spannweite an unterschiedlichen Geräten und Maschinen deutlich:

- Büros / Computerräume: u. a. Computer, Monitore, Tischleuchten, Ventilatoren,

- Haushaltsgeräte / Teeküche: u. a. Kühlschränke, Wasserkocher, Geschirrspülmaschinen, Mikrowellen, Kaffeemaschinen, Kaffeevollautomaten, Herde,
- Infrastruktur: Server, IT-Switches, Beleuchtungen,
- Werkzeugmaschinen: Spritzgussmaschinen, Oberfräsen, Drehmaschinen, u. v. m.

Anhand der Geräteausstattungen zeigen sich die unterschiedlichen Anforderungen: Im Privathaushalt bzw. in Büroumgebungen sind viele vorwiegend kleinere Geräte mit geringer Leistungsaufnahme zu finden. Im Fabrikationsumfeld sind im Vergleich nur wenige Maschinen jedoch mit erheblich höheren Leistungsaufnahmen installiert.

Auch die Kommunikationsanforderungen unterscheiden sich. Während IT-Geräte bereits eine Netzwerkanbindung besitzen, müssen andere Geräte erst kommunikationstechnisch angebunden werden. Für die Kommunikation mit Werkzeugmaschinen steht eine große Anzahl verschiedener Feldbusssysteme zur Verfügung. Viele dieser Feldbusse sind sogar echtzeitfähig. Für Haushaltsgeräte existieren bislang wenige offene Kommunikationsstandards. Deshalb wird hier auf Energiemessgeräte und Erkennungsalgorithmen zurückgegriffen.

### 6.1.3 Realisierte Energiemanagement-Anwendungen

Durch den Gesamtdemonstrator und seine Komponenten konnten die folgenden Energiemanagement-Anwendungen in den verschiedenen Umgebungen realisiert werden:

#### **Energiecockpit @ IWF, Lernfabrik / Systemweit**

Für die Umgebungen im IWF wurde das bereits in Unterabschnitt 5.3.1 vorgestellte Energiecockpit entwickelt. Auf einem großen Display werden permanent die Energieverbrauchswerte sowie viele weitere relevante Werte und Analysen angezeigt. Durch das Energiecockpit sind Nutzerinnen und Nutzer selbstständig in der Lage, spezifische Analysen (in Form von IoT-Softwareanwendungen) für die jeweiligen Maschinen zu aktivieren und zu konfigurieren.

Diese Anwendung wurde als Tablet- bzw. Smartphone-Anwendung auch in allen anderen Umgebungen installiert. So lassen sich damit auch die Serversysteme des IfN und die Geräte in den Büros des IBR analysieren.

#### **Meine Energie @ IfN, Teeküche**

Die Smartphone-Anwendung „Meine Energie“ fokussiert auf Echtzeitrückmeldungen der Aktivitäten der verwendeten Geräte und wurde für den Einsatz in Privathaushalten konzipiert (siehe Unterabschnitt 5.3.1). Für die Demonstration und Erprobung wurden die Geräte der Teeküche des IfN sowie einige Geräte in angrenzenden Büroräumen in der Anwendung als ein

gemeinsamer Haushalt dargestellt. Durch die Wiedergabe von aufgezeichneten Messdaten und durch gerätetechnische Nachbildungen wurden weitere Geräte als Simulation dem Demonstrationsaufbau hinzugefügt.

### Lastspitzenmanagement @ TUBS

Zur Demonstration der automatisierten Energiemanagement-Funktionen wurde ein Lastspitzenmanagement angewendet (siehe Unterabschnitt 5.3.3). Hierzu wurde die Struktur des Gesamtdemonstrators verwendet. Durch die Aufsummierung der Energieverbräuche der einzelnen Umgebungen, konnten reale und simulierte Geräte in Abhängigkeit von anderen Geräten gesteuert werden, ohne dass eine direkte Verknüpfung erforderlich war. Die Aufsummierung wurde jeweils in den einzelnen Instanzen der IoT-Softwareplattform durchgeführt und anschließend an die jeweils höhere Instanz vermittelt.

### Eigenverbrauchsoptimierung @ IfN

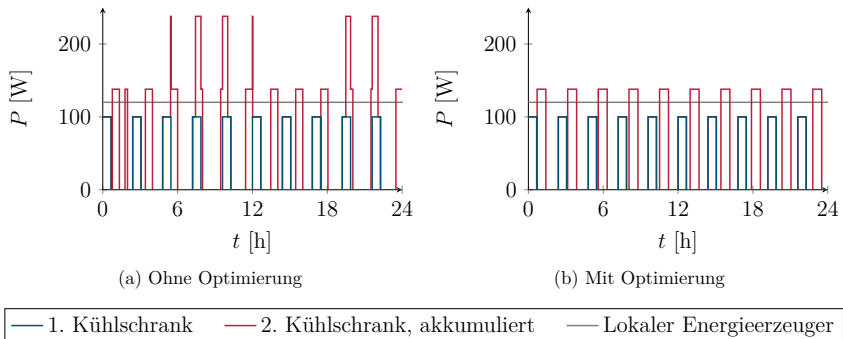


Abbildung 6.2: Prognose der Leistungsaufnahme zweier steuerbarer Kühlschränke (a) ohne Optimierung und (b) mit Optimierung. Abgebildet ist in Blau die Leistungsaufnahme des ersten Kühlschranks sowie in Rot die Aufsummierung beider Leistungsaufnahmen. Ziel der Optimierung ist die möglichst genaue Ausnutzung einer dauerhaften Erzeugungsleistung von 120 W eines fiktiven lokalen Energieerzeugers bei möglichst geringem Energiebezug aus dem Versorgungsnetz.

Die Eigenverbrauchsoptimierung für Prosumer aus Unterabschnitt 5.3.4 wurde mittels virtueller Geräte unter anderem im IfN erprobt. In Abbildung 6.2 ist ein vereinfachtes Beispiel dieser Optimierung dargestellt. Hier wird die Leistungsaufnahme zweier Kühlschränke mit der Erzeugungsleistung von 120 W eines fiktiven lokalen Energieerzeugers abgeglichen. Aus der Abbildung wird deutlich, dass sich die aufsummierte Leistungsaufnahme unter Verwendung der Optimierung besser an die lokale Erzeugung anpasst und weniger Energie aus dem Versorgungsnetz bezogen werden muss. Die Nutzung der lokal erzeugten Energie stieg in diesem vereinfachten Beispiel um 7,5 Prozentpunkte (von 83,8 % auf 91,3 %).

### Variable Tarife @ TUBS

Zur Erprobung von variablen Tarifen wurde im Koordinator „global“ eine Datenresource mit einem zufallsgenerierten Energiepreis veröffentlicht. Mittels der Anwendung em-real-time-pricing/event-price wurden einzelne Geräte des Demonstrators entsprechend angesteuert. Eine Verknüpfung mit den anderen Energiemanagement-Anwendungen erfolgte nicht.

### Direkteingriff im Sinne des Market Demand Response @ TUBS

Analog zur Erprobung der variablen Tarife wurde auch die Anwendung zum Direkteingriff im Sinne des Market Demand Response nach Unterabschnitt 5.3.2 eingesetzt. Die Anwendung market-dr/master wurde im Koordinator „global“ aktiviert. Einzelne Geräte wurden mittels der Anwendung market-dr/device dem Direkteingriff zugeordnet.

Auch der Einsatz der weiteren Anwendungen aus Abschnitt 5.3 wurde im Demonstrator demonstriert, so zum Beispiel die Anwendung event-device zur Abstimmung von Geräten untereinander, die Anwendung er-app-homecontrol zur Steuerung von Geräten durch Nutzerinnen und Nutzer sowie die Anwendungen zur automatisierten Abschaltung von Geräten aus Unterabschnitt 5.3.5.

## 6.1.4 Fazit

Aus den Beschreibungen in den vorangegangenen Unterabschnitten lässt sich ableiten:

- Dass das in dieser Arbeit implementierte System in verschiedenen Umgebungen und mit unterschiedlichen Verfahren zur Geräteeinbindung unter Einsatz verschiedener Funktionalitäten zum Energiemanagement einsetzbar ist.
- Ferner wird durch die Realisierung des Gesamtdemonstrators deutlich, dass Insellösungen (siehe Abschnitt 3.1) speziell nur für eine Art der Umgebung vermieden werden konnten. So sind einige existierende hoch spezialisierte Umgebungen (wie z. B. die Lernfabrik im IWF) mit der implementierten IoT-Softwareplattform ausgestattet und erfolgreich mit den restlichen Umgebungen verbunden worden.
- An dieser Stelle soll nochmals herausgestellt werden, dass es gelang, verschiedene Funktionalitäten mit unterschiedlichen Anforderungen in nur einem System zu implementieren. Dazu gehören das automatisierte Energiemanagement, das Energiemanagement mit Nutzereinbindung, aber auch Funktionalitäten jenseits des Energiemanagements. Dies stellt im Vergleich zu bestehenden Systemen eine besonders hervorzuhebende Eigenschaft dar.

- Zusätzlich wurden diverse Geräte mit Hilfe verschiedener Geräteeinbindungen bzw. unter Verwendung von verschiedenen Kommunikationsprotokollen verbunden. Darunter befanden sich sowohl nicht-vernetzte und auch vernetzte Geräte. Auch dieses ist im Vergleich zu vorhandenen Systemen hervorzuheben.

Die Eigenschaften der letzten beiden Teilabsätze wurden durch die gewählte Trennung in unterschiedliche Schichten (siehe Abschnitt 3.3) ermöglicht, wodurch Geräteeinbindungen und die Funktionalitäten zum Energiemanagement unabhängig voneinander realisiert werden konnten. Abschließend soll an dieser Stelle gefolgert werden, dass das in Kapitel 4 beschriebene Systemkonzept aus hierarchischer Kommunikationsstruktur, Geräteeinbindungen und Anwendungen mit der genannten Aufteilung in Schichten grundsätzlich umsetzbar ist.

## 6.2 Analyse des Designs und Verhaltens der IoT-Softwareplattform

Die gewählte hierarchische Kommunikationsstruktur (siehe Unterabschnitt 4.3.2) ist eine der wichtigsten Konzepte dieser Arbeit. Mit ihr wird das Ziel verwirklicht, Datenschutz- und Datensicherheitsanforderungen in verteilten Netzwerkstrukturen mit einer möglichst lokalen Verarbeitung der Daten erfüllen zu können. Ergänzend dazu ist die IoT-Softwareplattform als Kernkomponente für den Einsatz in jedem Knoten der hierarchischen Kommunikationsstruktur entwickelt worden (siehe Abschnitt 5.1). Anhand der IoT-Softwareplattform wird in diesem Abschnitt unter anderem diskutiert, welche Vor- und Nachteile die Verwendung der hierarchischen Kommunikationsstruktur aufweist.

In Unterabschnitt 6.2.1 wird zunächst das Systemverhalten bei der Verwendung der IoT-Softwareplattform analysiert. Dabei wird insbesondere die Leistungsfähigkeit der Datenvermittlung untersucht. In Unterabschnitt 6.2.2 werden anschließend die Latenz und die erzeugten Datenraten in der Kommunikationsstruktur im Vergleich zu den in Unterabschnitt 2.3.5 aufgestellten Systemtypen analysiert.

In Unterabschnitt 6.2.3 und 6.2.4 wird anschließend überprüft, ob und auf welche Weise die funktionalen und nicht-funktionalen Anforderungen bei der Konzeption und Implementierung beachtet wurden. Dabei werden die konkreten Lösungen sowie die zugrundeliegenden Konzepte betrachtet und bewertet.

Abschließend wird in Unterabschnitt 6.2.5 ein Fazit über die Verwendung der hierarchischen Kommunikationsstruktur und der IoT-Softwareplattform gezogen.

## 6.2.1 Evaluation des Systemverhaltens

Auf Basis der IoT-Softwareplattform können Anwendungen im Sinne der hierarchischen Kommunikationsstruktur über mehrere Geräte verteilt ausgeführt werden. Dazu müssen Daten über mehrere Instanzen hinweg kommuniziert werden. Für die Kommunikation werden ausschließlich die in Abschnitt 5.1 beschriebenen Basisfunktionen verwendet. In diesem Unterabschnitt soll zunächst die Leistungsfähigkeit dieser Basisfunktionen und der vorgenommenen Implementierung betrachtet werden. Für die Kommunikation können RPC-Aufrufe (siehe Methoden `rpc` und `on_rpc` in Unterabschnitt 5.1.3) oder das Publish-Subscribe-Entwurfsmuster verwendet werden. Üblicherweise wird das Publish-Subscribe-Entwurfsmuster (siehe Methoden `publish` und `subscribe` in Unterabschnitt 5.1.2) verwendet, wenn Daten *von* einer Datenressource zu einem oder mehreren fremden Abonnenten übermittelt werden sollen. Hingegen werden RPC-Aufrufe verwendet, wenn Daten *zu* einer spezifischen Datenressource übermittelt werden sollen. Bei diesen Daten handelt es sich in der Regel um Steuerungsbefehle.

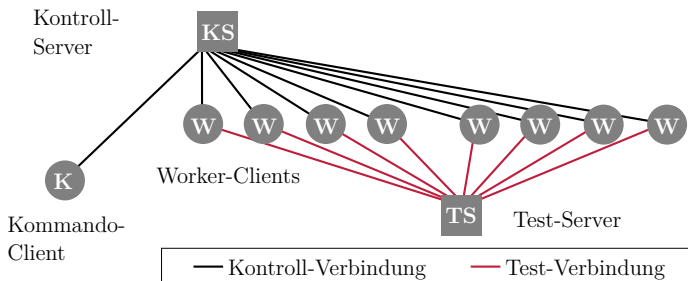


Abbildung 6.3: Struktur des Evaluationsframeworks, bestehend aus einem Kontroll-Server (KS) mit Worker-Clients (W), einem Kommando-Client (K), der Aufgaben über den Kontrollserver an die Worker-Clients sendet und einem Test-Server (TS), der von den Worker-Clients angesprochen wird. Das Framework gestattet es, mittels vieler Worker-Clients den Test-Server anzusprechen und seine Leistungsfähigkeit zu protokollieren.

Zur Evaluation dieser Netzwerk-Funktionen der IoT-Softwareplattform wurde ein verteiltes Evaluationsframework ebenfalls auf Basis der IoT-Softwareplattform realisiert. Die Struktur des Evaluationsframeworks ist in Abbildung 6.3 dargestellt. Ziel ist die Erfassung von Paketumlaufzeiten bei verschiedenen Belastungssituationen. Dazu wird ein Test-Server (TS) parallel von vielen Clients, hier Worker-Clients (W) angesprochen. Die Belastungssituationen werden durch die Variation der Anzahl an Anfragen durch die Clients an den Server geschaffen. Dazu wird eine Struktur aus einem Kontroll-Server (KS) und einem Kommando-Client (K) verwendet, die Aufgaben an die Worker-Clients sendet. Die Aufgaben werden

als Anwendungskonfigurationen definiert. Sie werden von den Worker-Clients automatisch ausgeführt und ihre Ergebnisse zurück an den Kommando-Client gesendet.

Alle Testprozeduren wurden auf einem Server, Baujahr 2012 mit acht Prozessorkernen unter Debian Linux 9.8 und der Node.js-Version 9.11 durchgeführt.

### Latenz (RPC)

Die *Latenz* beschreibt in der Kommunikationstechnik die Verzögerungszeit, in der ein Paket übertragen wird (siehe auch Unterabschnitt 3.3.4). Diese Größe kann über die Messung der Paketumlaufzeit bestimmt werden. Dazu wird eine Anfrage gesendet und die Zeit bis zu einer Rückantwort ermittelt. Die Latenz entspricht dabei der halben Paketumlaufzeit.

Im Evaluationsframework werden die Latenzen in den Worker-Clients erfasst, die Anfragen an den Test-Server senden. Aufgrund dieser zusätzlichen Messung und Berechnung, benötigt ein einzelner Client mehr Rechenressourcen als die Instanz des Test-Servers. Um den Server auszulasten, werden mehrere parallele Clients benötigt. Damit die Anfragen den Test-Server nicht alle synchron erreichen, werden die Aufgaben auf den Test-Clients mit einer geringen zufälligen Verzögerung gestartet. Die gemessenen Werte werden über ein Intervall von 20 Sekunden gemittelt.

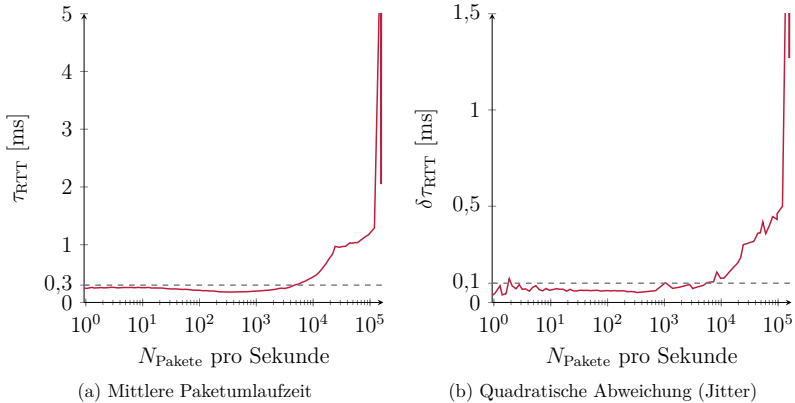


Abbildung 6.4: Darstellung der (a) mittleren Paketumlaufzeit und (b) deren mittlere quadratische Abweichung bei einem RPC-Aufruf zwischen zwei lokalen Koordinatoren aufgetragen gegen die Anzahl der übermittelten Pakete.

Die Abbildung 6.4a zeigt die gemittelte Umlaufzeit eines RPC-Aufrufs mit Antwort zwischen zwei lokalen Koordinatoren in Abhängigkeit zur Anzahl der übermittelten Pakete. Mit über 170.000 Datensätzen pro Sekunde und Server-Instanz konnten wesentlich mehr Daten beantwortet werden, als im Gesamtdemonstrator (siehe Abschnitt 6.1) in den lokalen Umgebungen

anfielen. Damit ist die Implementierung auch für den Praxiseinsatz geeignet und kann auch im Vergleich mit weitverbreiteten quelloffenen Softwareprodukten zur Datenvermittlung bestehen (vgl. z. B. [Sca15, S. 9]).

Bei genauer Betrachtung der Abbildung 6.4a wird deutlich, dass die Werte im Bereich von 0 bis zu 300 Paketen pro Sekunde leicht abfallen. Die ursprüngliche Annahme, dass dies an der Anwendung des sogenannten Nagle-Algorithmus liegt, kann ausgeschlossen werden, da dieser Algorithmus bei der verwendeten WebSocket-Implementierung bei der Verbindungsinitialisierung abgeschaltet wird. Beim Nagle-Algorithmus handelt es sich um eine TCP-Optimierung im Betriebssystem bei der kleine TCP-Pakete automatisch zusammengefasst werden. Es ist deshalb anzunehmen, dass die höheren Paketumlaufzeiten bei geringem Paketaufkommen auf die Reaktivierungszeit der Anwendung im Betriebssystem zurückzuführen sind. Mit steigender Paketanzahl ist die Server-Anwendung mit einer höheren Wahrscheinlichkeit noch aktiv und kann damit die Pakete direkt verarbeiten.

Die Implementierung des Protokolls in der IoT-Softwareplattform fasst Pakete zusammen, die direkt nacheinander gesendet werden. Diese Optimierung ist in der Abbildung ab einer Anzahl von etwa 25.000 Datensätzen pro Sekunde erkennbar. Durch diese Optimierung erhöht sich die Umlaufzeit, jedoch ist der Server weiterhin in der Lage die große Anzahl der Pakete zu verarbeiten.

Auch bei hohen Lastsituationen mit etwa 150.000 Datensätzen pro Sekunde gehen keine Pakete verloren, da das Protokoll indirekt auf TCP/IP aufsetzt. Durch die Implementierung werden ebenfalls keine Pakete verworfen, da die Pakete zwischengespeichert und nach und nach bearbeitet werden. Wie in der Abbildung 6.4a deutlich wird, erhöhen sich dadurch aber die Paketumlaufzeiten so deutlich, dass solche nicht wünschenswerten Lastsituationen des Servers vermieden werden sollten.

### **Jitter (RPC)**

Als *Jitter* wird die Abweichung von der mittleren Latenz bezeichnet. Ein zu hoher Wert kann zu Problemen bei der Ausführung der Anwendungen führen. Einige Anwendungen benötigen möglichst gleichbleibende Latenzen, um fehlerfrei zu arbeiten, so zum Beispiel Telefonie- und Gaming-Anwendungen. Zur Bestimmung des Jitters finden sich verschiedene Definitionen. Viele basieren auf periodisch versendeten Paketen, die in dieser Evaluation jedoch nicht auftreten. Deshalb wird hier für den Jitter die mittlere quadratische Abweichung von der mittleren Umlaufzeit betrachtet.

In Abbildung 6.4b ist diese Abweichung in Abhängigkeit von der Anzahl der Datenpakete dargestellt. Mit Abweichungen unter 0,1 ms bei bis zu 6.000 Datensätzen pro Sekunde (siehe rote Linie) und bis zu 0,5 ms bei bis zu 100.000 Datensätzen pro Sekunde kann gefolgert



werden, dass die Abweichung von der Latenz so gering ist, dass keine störenden Effekte zu erwarten sind.

### Latenz (Publish-Subscribe)

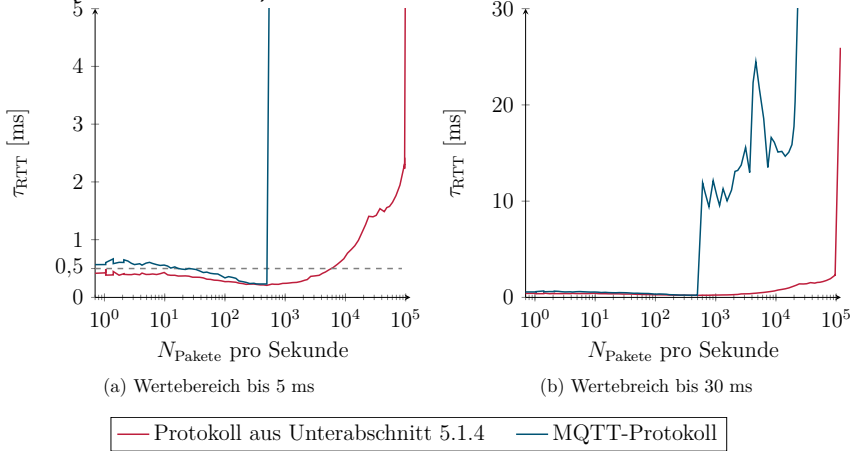


Abbildung 6.5: Mittlere Paketumlaufzeit zwischen zwei lokalen Koordinatoren unter Nutzung des Publish-Subscribe-Entwurfsmusters aufgetragen gegen die Anzahl der übermittelten Pakete. In Rot unter Nutzung des Protokolls aus Unterabschnitt 5.1.4, in Blau unter Nutzung des MQTT-Protokolls (siehe Unterabschnitt 2.3.4). (a) und (b) zeigen verschiedene Wertebereiche der gleichen Daten.

In Abbildung 6.5 wird analog zur Latenz bei der Nutzung von RPC-Methoden, die Latenz bei der Nutzung der Funktionen `publish` und `subscribe` über eine entfernte Instanz aufgetragen. Dabei ist festzustellen, dass sich dieses Entwurfsmuster ähnlich performant wie die Nutzung der RPC-Methoden verhält. So können bis zu 120.000 Ereignisse übertragen werden. Im direkten Vergleich weist die Nutzung der RPC-Methoden eine bessere Performance auf: Die Latenzen sind etwas geringer und bei voller Auslastung kann eine höhere Anzahl an Anfragen verarbeitet werden. Die etwas geringere Performance war zu erwarten, da das Publish-Subscribe-Entwurfsmuster im Vergleich zu den RPC-Aufrufen etwas aufwändiger ist, weil prinzipiell mehrere Abonnenten eine Datenressource abonnieren können.

Zusätzlich ist in Abbildung 6.5 die Latenz bei der Nutzung des MQTT-Protokolls aus Unterabschnitt 2.3.4 aufgetragen. Mittels des Softwaremoduls MQTT.js wurde eine Anwendung zur Kommunikation mit einem MQTT-Server für die IoT-Softwareplattform geschaffen. Für die Serverinstanz wurde die Software mosquitto verwendet. Als Dienstgüte wurde  $QoS = 2$  eingestellt, da diese dem Protokoll aus Unterabschnitt 5.1.4 am Besten entspricht. Aus dem Vergleich der Latenzen wird erneut die Performance der IoT-Softwareplattform deutlich.

Dabei muss die unterschiedliche Achsenskalierung beachtet werden, ohne die keine annehmbare Darstellung möglich gewesen wäre. Mit dem MQTT-Protokoll können entsprechend bis zu 20.000 Ereignisse bei erblich schlechteren Latenzen übertragen werden.

## 6.2.2 Systemverhalten unter Verwendung der hierarchischen Kommunikationsstruktur

In Erweiterung zum vorhergehenden Unterabschnitt 6.2.1 sollen nachfolgend die Eigenschaften und das Verhalten der IoT-Softwareplattform in der hierarchischen Kommunikationsstruktur betrachtet werden. Die Kommunikationsstruktur soll hierbei insbesondere im Vergleich zu den Systemtypen 0 bis 2 aus Unterabschnitt 2.3.5 betrachtet werden. Hilfreich dabei ist, dass mit Hilfe der IoT-Softwareplattform auch Kommunikationsstrukturen nach diesen Systemtypen realisiert werden können.

Damit die hierarchische Kommunikationsstruktur sinnvoll eingesetzt werden kann, müssen zusätzliche lokale Koordinatoren in der Kommunikation eingebracht werden. Damit ist die Frage zu klären, welche Effekte diese zusätzliche Einbringung von Instanzen der IoT-Softwareplattform bewirkt. Aufgrund des sich ergebenden komplexeren Aufbaus bei Verwendung der hierarchischen Kommunikationsstruktur kann vermutet werden, dass sich Paketlaufzeiten und -datenraten im Vergleich zu bestehenden Systemen verschlechtern. Dieser Fragestellung soll in diesem Unterabschnitt nachgegangen werden. Zunächst wird dazu die zusätzliche Latenz abschätzt, die durch das Einbringen eines zusätzlichen lokalen Koordinators entsteht. Anschließend werden in einem Testszenario die Systemtypen mit der hierarchischen Kommunikationsstruktur verglichen.

### Latenz in der hierarchischen Kommunikationsstruktur

Wird der gleiche Kommunikationsweg verwendet, muss aufgrund der zusätzlichen lokalen Koordinatoren mit einer höheren Latenz gerechnet werden. Jeder Koordinator im Kommunikationsweg wird hier als ein sogenannter *Hop* bezeichnet. Ein solcher zusätzlicher Hop erzeugt

$$\Delta t_{\text{Hop}} = \frac{\tau_{\text{RTT}}}{2} + \Delta t_{\text{zusätzliche Netzwerkstrecke}}$$

an zusätzlicher Latenz pro Wegstrecke. Sofern die Netzwerkpakete keine zusätzlichen (oder andere) Wege nehmen, ist mit einer zusätzlichen Verzögerung von ungefähr  $\Delta t_{\text{RPC,Hop}} = 0,15$  ms pro Hop bei RPC-Methoden und von ungefähr  $\Delta t_{\text{PS,Hop}} = 0,25$  ms bei Verwendung der Publish-Subscribe-Methoden zu rechnen (siehe gestrichelte Linien in der Abbildung 6.4a sowie in der Abbildung 6.5). Dabei wird vorausgesetzt, dass die jeweilige Instanz nicht überlastet ist.

Im Vergleich dazu betragen die Paketantwortzeiten im Internet z. B. zwischen Frankfurt und New York mindestens 41,4 ms oder zwischen Braunschweig und Frankfurt mindestens 2,34 ms.<sup>3</sup> Damit ist die oben ermittelte Größe auch bei mehreren Hops vernachlässigbar.

Durch die Nutzung der hierarchischen Kommunikationsstruktur kann sich sogar ein Latenzgewinn ergeben, da vielfach wesentlich kürzere Kommunikationsstrecken verwendet werden. Anforderungen an die Latenz finden sich überwiegend zwischen benachbarten Objekten, die miteinander interagieren sollen. Ein intelligenter Lichtschalter soll das Licht und ggfs. weitere Geräte einschalten, ohne dass die Nutzerin bzw. der Nutzer eine wesentliche Verzögerung bemerkt. Auch wenn zwei Werkzeugmaschinen kooperieren sollen, ergeben sich Latenz- oder sogar Echtzeitanforderungen. Durch einen lokalen Koordinator kann die Kommunikation über einen reinen lokalen Kommunikationsweg erfolgen und dadurch erheblich kleinere Latenzen realisiert werden. Dies ist insbesondere im Vergleich mit Cloud-basierten Systemen des Systemtyps 2 (siehe Unterabschnitt 2.3.5) zu betrachten, bei denen diese Kommunikationswege unabhängig von der Entfernung in der Regel über die Cloud oder ein anderes zentrales System verlaufen.

Eine strenge Echtzeitfähigkeit kann hingegen auf Basis der IoT-Softwareplattform nicht garantiert werden, da das für die Kommunikation verwendete Internet Protokoll (IP) hierfür keine Mechanismen besitzt.

### **Datenraten in verteilten Strukturen**

Im folgendem Testszenario wird von der Annahme ausgegangen, dass der Energieverbrauch über einhundert Haushalte mit jeweils einhundert Geräten aggregiert werden soll. Insgesamt sollen also zehntausend Geräte verbunden werden. Für die Aggregation wird von jedem Gerät jede Sekunde eine Nachricht mit einer durchschnittlichen Größe von 104 Byte versendet. Die Paketgröße ist nicht repräsentativ, da der vom Nutzer gewählte Gerätenamen in jedem Paket enthalten ist.

Zur weiteren detaillierten Betrachtung werden zwei Fälle angenommen, die in Abbildung 6.6 verdeutlicht sind. Die Anzahlen der jeweiligen Geräte sind in der Abbildung jeweils rechts neben den Gerätesymbolen angegeben. Im Fall (a) werden alle Geräte direkt an den zentralen Koordinator angeschlossen. Dieser Fall entspricht dem Systemtyp 2 aus Unterabschnitt 2.3.5. Im Fall (b) werden zusätzlich weitere 100 lokale Koordinatoren in die Haushalte eingebracht und an den zentralen Koordinator angeschlossen. Dabei werden die untergeordneten Geräte an die jeweiligen lokalen Koordinatoren angeschlossen. In den nun zwischengeschalteten lokalen Koordinatoren wird jeweils der Energieverbrauch aller angeschlossenen Geräte

---

<sup>3</sup>Diese Latenzen wurden aus der Multiplikation des zweifachen direkten Abstands und der Lichtgeschwindigkeit bestimmt.

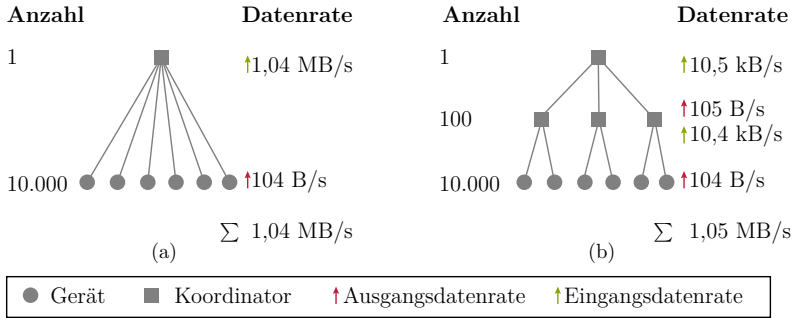


Abbildung 6.6: Darstellung der Datenraten in zwei Fällen: Unter (a) werden alle Geräte direkt mit einem zentralen Koordinator verbunden. Unter (b) werden zusätzlich lokale Koordinatoren zwischengeschaltet. Links sind jeweils die Anzahl der Entitäten dargestellt und rechts die Eingangs- bzw. Ausgangsdatenraten.

aggregiert und an den zentralen Koordinator vermittelt. Dabei werden sekundlich Pakete mit einer durchschnittlichen Größe von 105 Byte generiert. Auch diese Paketgröße ist nicht repräsentativ. Dieser Fall entspricht dem Systemtyp 1 aus Unterabschnitt 2.3.5.

Die angenommenen Datenraten, die von jedem einzelnen Gerät gesendet bzw. empfangen wurden, sind in Abbildung 6.6 rechts neben den Gerätesymbolen abgebildet. Im Vergleich der beiden Fälle reduziert sich am zentralen Koordinator für den Fall (b) die Datenrate auf einen Wert von nur noch 1,01 %. Auch die Upload-Datenrate jedes Haushalts sinkt von 10,4 kB/s auf 105 B/s. Wie aus der Aufsummierung in der Abbildung ersichtlich, steigt dabei gleichzeitig die Anzahl der insgesamt übertragenen Datenpakete. Diese Pakete treffen jedoch anders als im ersten Fall in keinem Punkt des Netzwerkes zusammen.

Wird eine dritte Ebene eingefügt, könnte bereits der Energieverbrauch über eine Millionen Geräte aggregiert werden. Dazu würden jedoch auch 10.101 lokale Koordinatoren benötigt. Diese Struktur entspricht keinem der bisherigen Systemtypen.

Neben der deutlichen Senkung der Datenraten wird dabei deutlich, dass sich verschiedenste Netzwerkstrukturen auf Basis der IoT-Softwareplattform realisieren lassen. Im Vergleich zum Systemtyp 1 lassen sich z.B. weitere lokale Koordinatoren in Ebenen über- oder unterhalb des lokalen Servers einsetzen. So lassen sich lokale Umgebungen weiter unterteilen oder mehrere lokale Umgebungen zusammenschließen. Insbesondere durch die weitere Unterteilung von lokalen Umgebungen können sich Vorteile, wie geringere Latenzen und höhere Übertragungskapazitäten ergeben.

### 6.2.3 Umsetzung der funktionalen Anforderungen

Das Design der Implementierung wird nachfolgend anhand der Anforderungen F.1 bis F.6 aus Unterabschnitt 3.3.3 diskutiert. Dabei wird herausgestellt, welche konkreten Lösungen zur Erfüllung der aufgestellten Anforderungen gewählt wurden.

#### Kapselung der Funktionen nach dem Schichtenmodell

Jedes Teilsystem basiert auf der IoT-Softwareplattform. Wie in Abschnitt 5.1 erläutert, wird die IoT-Softwareplattform durch Anwendungen erweitert. Damit wird die Kapselung der Funktionen realisiert, die in **Anforderung F.1** gefordert wird. Durch die Kapselung sollen sowohl Funktionen der Schicht der Geräteeinbindung als auch der Schicht der Funktionalitäten in die IoT-Softwareplattform eingebunden werden.

Um eine möglichst hohe Flexibilität zu erreichen, wurde keine strikte Aufteilung in verschiedene Anwendungstypen gewählt. Alle Anwendungen können auf die gleiche Weise geladen werden und alle Funktionen nutzen. Damit ist dennoch die Möglichkeit gegeben, Anwendungen zu implementieren, die gleichzeitig Funktionen beider Schichten realisieren.

Unter Betrachtung der realisierten Anwendungen wird deutlich, dass der Großteil der implementierten Anwendungen eindeutig einer der genannten Schichten zugeordnet werden kann. Die Anwendungen aus Unterabschnitt 5.1.7 bis 5.1.9 sowie Abschnitt 5.2 dienen der Geräteeinbindung. Die Anwendungen aus Abschnitt 5.3 implementieren die Schicht der Funktionalitäten. Hilfs- und Administrationsanwendungen sind entweder der Schicht der Kommunikation oder keiner der drei Schichten zuzuordnen.

Durch die Aufteilung und Kapselung der Funktionen ist es möglich, verschiedene Anwendungen frei miteinander zu kombinieren. Dies wird am Gesamtdemonstrator deutlich, wo verschiedene Verfahren zur Geräteeinbindung verwendet werden und die Grundlage für die gleichen Funktionalitäten bilden.

Mit Blick auf die Anforderung F.1 kann gefolgert werden, dass die Kapselung der Funktionen nach dem Schichtenmodell möglich und auch sinnvoll ist. Dennoch erwies es sich als zweckdienlich, keine Beschränkungen in der Implementierung vorzugeben und auch andere Anwendungstypen zuzulassen. Die Anwendungen können individuell pro Teilsystem installiert, geladen und konfiguriert werden – wodurch auch der zweite Teil der Anforderung erfüllt ist.

#### Transparente Kommunikation

Die IoT-Softwareplattform verbindet die Anwendungen untereinander und ist deshalb der Schicht der Kommunikation zuzuordnen. Die Kommunikation zwischen den Anwendungen

erfolgt ausschließlich über die Datenressourcen (siehe Unterabschnitt 5.1.1). Mit Hilfe der IoT-Softwareplattformen werden die von den Anwendungen erzeugten Daten in den Datenressourcen gespeichert und können wiederum von anderen Anwendungen abonniert werden. Dabei können sowohl lokal erzeugte, wie auch entfernte Datenressourcen angesprochen werden. Auch der Austausch der Datenressourcen über mehrere Instanzen ist möglich, sofern die zugrundeliegenden Kommunikationsverbindungen zwischen den Instanzen aufgebaut und alle notwendigen Berechtigungen eingeräumt wurden.

Aus Sicht der Anwendungen wird damit die komplette Kommunikation – wie in **Anforderung F.2** gefordert – transparent ausgeführt, da die Anwendungen die Kommunikation weder beachten noch steuern müssen. Eine Ansprache bestimmter Netzwerkgeräte ist weder vorgesehen noch möglich. Alle Schnittstellen (siehe Unterabschnitt 5.1.2 und 5.1.3) enthalten deshalb keine Argumente, um die Kommunikation zu steuern.

Wie weiterhin in Anforderung F.2 gefordert, können mit Hilfe der IoT-Softwareplattform unterschiedlichste Datenzugriffe realisiert werden. Damit wird es möglich, dass prinzipiell alle Komponenten miteinander kommunizieren können. Konkret werden in Anforderung F.2 vier Grundtypen an Verbindungen gefordert, die nachfolgend diskutiert werden:

- Anwendungen können auf die Daten von anderen Anwendungen zugreifen, indem sie deren Datenressource abonnieren (siehe Unterabschnitt 5.1.2). Da die Verfahren zur Geräteeinbindung als Anwendungen implementiert wurden, ist damit auch der Zugriff auf Gerätedaten abgedeckt.
- Auch eine bidirektionale Kommunikation zwischen einzelnen Anwendungen ist über die Schnittstelle zur Definition und zum Aufruf von RPC-Methoden möglich (siehe Unterabschnitt 5.1.3). Mit dieser Funktion wird auch die Beeinflussung von Geräten durch andere Anwendungen realisiert.
- Zur Kooperation zwischen den Anwendungen können sowohl die Möglichkeit eines Abonnements von Daten als auch der Aufruf der RPC-Funktionen anderer Anwendungen genutzt werden. Durch die RPC-Funktionen kann zudem eine bidirektionale Kommunikation zwischen den Komponenten realisiert werden.
- Globale Informationen oder Vorgaben können direkt in einer separaten Anwendung gehalten werden. Auch eine Einbindung externer Datenquellen durch eine Anwendung ist möglich. Der Koordinator „global“ des Gesamtdemonstrators (siehe Abschnitt 6.1) hält so z. B. die angestrebte Leistungsaufnahme für die Berechnung der Lastspitzenreduktion.

Alle benannten Verbindungsarten wurden für die Umsetzung des Gesamtdemonstrator „Real-Life-Lab Campus 2020“ (siehe Unterabschnitt 6.1.1) verwendet.

### **Latenzanforderungen für die Kommunikation**

Bereits in Unterabschnitt 6.2.1 wurde dargestellt, wie sich die Latenz in Abhängigkeit verschiedener Einflussgrößen verhält. An dieser Stelle soll diskutiert werden, welche Maßnahmen zur Minimierung der Latenz getroffen wurden.

Das zugrunde gelegte Publish-Subscribe-Entwurfsmuster (siehe Unterabschnitt 2.3.3 bzw. Unterabschnitt 5.1.2) ermöglicht eine direkte Weiterleitung von neuen Informationen an interessierte Abonnenten. Im Vergleich zu anderen Techniken entstehen hier keine zusätzlichen Verzögerungen. Hier sei beispielsweise das *Polling* (engl. für Abfragen) genannt, das für das zyklische Abfragen einer Ressource steht. Beim Polling tritt eine zusätzliche Verzögerung zwischen der Veröffentlichung und der nächsten zyklischen Abfrage auf.

Die RPC-Funktionen ermöglichen die direkte gezielte Ansteuerung einer Ressource. Informationen werden dadurch auf direktestem Weg (unter Einhaltung der zugrunde liegenden Topologie) an genau einen Empfänger vermittelt. Da hier keine zeitverzögernden Techniken, wie das genannte Polling, zum Einsatz kommen, entstehen dabei keine weiteren Verzögerungen.

Für die Kommunikation zwischen zwei Instanzen wird eine dauerhafte aktive WebSocket-Verbindung verwendet. Im Gegensatz zu anderen Protokollen, wie REST, MQTT und CoAP, wird aufgrund der (bereits) aktiven Verbindung keine zusätzliche Verzögerung für den Verbindungsaufbau zwischen den Instanzen benötigt. Die aktiven Verbindungen führen allerdings insbesondere bei geringen Datendurchsätzen zu einem zusätzlichen Netzwerkverkehr. Der Overhead, also die nicht zu den Nutzdaten zählenden, übertragenen Daten, steigt an.

Zusammenfassend wird deutlich, dass die Reaktionszeiten innerhalb der IoT-Softwareplattform, wie in **Anforderung F.3** gefordert, so gering wie möglich gehalten werden können. Wie in Unterabschnitt 6.2.3 gezeigt, können bei direkter lokaler Kommunikation die Verzögerungen auf Werte unterhalb von 0,15ms bei RPC-Methoden und 0,25ms bei Verwendung der Publish-Subscribe-Methoden reduziert werden. Dazu wurde der Einsatz von Techniken vermieden, die zusätzliche Verzögerungen verursachen könnten. Um die Latenzanforderungen konkreter Funktionalitäten erfüllen zu können, müssen folglich vor allem die Netzwerkverbindungen zwischen den Geräten sowie die Verzögerungen durch Berechnungsalgorithmen und fremde Kommunikationstechniken beachtet werden.

### **Begrenzung der Kommunikation / Zugriffsschutz**

Aufgrund der dargestellten vielfältigen Kommunikationsmöglichkeiten und der übergreifenden Anwendungsfelder sind Schutzmaßnahmen erforderlich, um die Daten und Funktionen der Datenressourcen vor unberechtigtem, fremdem Zugriff zu schützen. Dies wird durch die hierarchische Kommunikationsstruktur (siehe Unterabschnitt 4.3.2) und das gewählte Grundkonzept der lokalen Verarbeitung von Daten maßgeblich unterstützt. Daten müssen dabei nicht weiter als notwendig vermittelt werden.

Durch die Kommunikationsstruktur können Systemgrenzen zwischen den Umgebungen eingezogen werden, an denen Daten- und Funktionszugriffe eingeschränkt werden können. Anhand des Gesamtdemonstrators wird eine solche Aufteilung in Teilsysteme in Abbildung 6.1 verdeutlicht. Die Einschränkungen werden anhand von Regelsätzen überprüft (siehe Unterabschnitt 5.1.5), die durch die Administratorinnen und Administratoren sowie auch durch installierte Anwendungen vorgegeben werden können. Dadurch wird eine feingranular gestaffelte Weitergabe der Datenressourcen z. B. durch Einschränkung oder vollständige Unterbindung der Datenweitergabe an höhere Instanzen möglich. Die Regelsätze werden auch beim Datenzugriff von Anwendungen herangezogen. Dadurch ist es wiederum möglich, einzelne Anwendungen in ihrem Datenzugriff einzuschränken.

Zusätzlich bietet die IoT-Softwareplattform Möglichkeiten zur Authentifizierung von Kommunikationspartnern an. Dadurch können unterschiedliche Instanzen voneinander unterschieden und passende Regelsätze angewendet werden.

Damit ist die **Anforderung F.4 erfüllt**, da alle Kommunikationswege durch die IoT-Softwareplattform, also sowohl zwischen den Instanzen und auch zwischen den Anwendungen, individuell und feingranular in ihrer Datenweitergabe beschränkt werden können.

### **Konfiguration durch automatische Verknüpfung / Auto Discovery**

Bisher wurde dargelegt, dass die Kommunikationsfähigkeit zwischen allen Ressourcen (wie Geräten und Funktionalitäten) gegeben ist. In **Anforderung F.5** wird gefordert, dass die Verknüpfung der Kommunikationspartner vorzugsweise automatisch erfolgen soll. Um diese Forderung zu erfüllen, müssen sich die Anwendungen in der IoT-Softwareplattform selbst identifizieren und entscheiden können, welche Datenressourcen von anderen Anwendungen für sie von Interesse sind.

Dies wird über die Funktion `subscribe_announcements` realisiert, die einer Anwendung alle Datenressourcen eines Bereiches mit ihren Metadaten zur Verfügung stellt (siehe Unterabschnitt 5.1.1 und 5.1.2). Die Metadaten sind als Schlüssel-Wert-Paare ausgeführt. Dadurch können die Anwendungen entweder anhand der Prüfung auf Existenz eines Schlüssels oder



anhand eines Wertes eines Schlüssels die Entscheidung treffen, ob die Daten der Datenressource für die Bewältigung ihrer Aufgaben erforderlich sind. Anhand der Metadaten { "Typ": "Temperatur", "Außentemperatur": true } könnte beispielsweise eine Anwendung in Form eines Heizungsreglers automatisch einen benachbarten Sensor für die Außentemperatur auffinden. Damit sich zwei Kommunikationspartner identifizieren können, ist es erforderlich, dass bei der Veröffentlichung und Zuordnung die gleichen Metadaten-Schlüssel und ggfs. auch -Werte verwendet werden. Wie in Unterabschnitt 5.1.1 erläutert, gibt es dazu verschiedene Initiativen, die eine Vereinheitlichung solcher Metadaten vorantreiben. Eine vollständige Vereinheitlichung muss hier jedoch nicht erreicht werden, da die Anwendungen auch mehrfach unterschiedliche Metadaten-Einträge prüfen können.

Durch diese Zuordnung, die in der Version in Abschnitt 3.1 auch als Auto Discovery bezeichnet wird, sind Anwendungen in der Lage ihren Lösungsraum selbstständig zu erweitern und weitere Daten für sich gewinnbringend zu nutzen. In Kombination mit der Kommunikationsfähigkeit der Anwendungen und Geräte wird damit die automatische Mehrfachnutzung von Sensoren erreicht, die eine der Grundideen des IoT darstellt.

### Nutzeroberflächen

Wie in **Anforderung F.6** gefordert, können Anwendungen über das Framework zur Darstellung von Nutzerschnittstellen eigenständige Ansichten für Nutzerinnen und Nutzer aufbereiten und anzeigen lassen. Wie in Unterabschnitt 5.1.6 beschrieben, wird auf das Model-View-ViewModel-Entwurfsmuster zurückgegriffen. Für die wichtigsten Basismethoden (siehe Unterabschnitt 5.1.2) wurden zusätzlich vergleichbare ViewModels bereitgestellt, die eine direkte Anzeige der über die Methoden bereitgestellten Daten ermöglichen. Dabei werden Datenaktualisierungen und weitere Daten unmittelbar zur Anzeige gebracht. Die Ansichten können auf Smartphones, Tablets, eigenständigen Displays sowie Laptops und Desktop-Computern dargestellt werden.

Hervorzuheben ist, dass das Framework auf der IoT-Softwareplattform basiert. So können viele Anwendungen, die für die IoT-Softwareplattform entwickelt wurden, auch in diesem Framework ausgeführt werden. Damit ist es möglich, Funktionalitäten vollständig, inkl. der Nutzeroberfläche, auf Basis der IoT-Softwareplattform umzusetzen.

## 6.2.4 Umsetzung der nichtfunktionalen Anforderungen

An der realisierten Struktur des Gesamtdemonstrators (siehe Abschnitt 6.1) wird deutlich, dass die Umsetzung einer hierarchischen Kommunikationsstruktur auch in der Praxis problemlos möglich ist. Dabei war der zusätzliche Aufwand, separate Geräte bzw. Teilsysteme

aufzusetzen, beim Aufbau des Gesamtdemonstrators zu vernachlässigen, da die Installation der eingesetzten Mini-Computer sowie virtuellen Server automatisiert werden konnte. Weiterhin erbringt die separat und zeitversetzt umgesetzte Aufteilung in mehrere Teilsysteme folgende administrative Vorteile:

- Die Teilsysteme unterliegen durch ihre unterschiedlichen Ausprägungen und Umgebungen verschiedenen Anforderungen, die durch die separaten Systeme individuell adressiert werden können,
- Wartung, Aktualisierung und Erweiterung können pro Teilsystem durchgeführt werden,
- Das System kann langsam organisch wachsen, ohne dass an anderer Stelle zusätzliche Ressourcen benötigt werden.

Diese und weitere Punkte werden nachfolgend anhand der nichtfunktionalen Anforderungen NF.1 bis NF.7 aus Unterabschnitt 3.3.3 diskutiert.

### Skalierungsfähigkeit

Aufgrund der potentiellen Größe des Systems wird in **Anforderung NF.1** eine gute Skalierungsfähigkeit insbesondere im Vergleich zu zentralisierten Systemen gefordert.

Da Daten lokal verarbeitet oder durch geeignete Aggregationen nur einzelne zusammengefasste Daten in der Struktur weitergeleitet werden müssen, bietet die hierarchische Kommunikationsstruktur auch bei der Skalierungsfähigkeit Vorteile. Bei Erweiterungen des Systems durch neue Umgebungen müssen auch zusätzliche lokale Koordinatoren installiert werden. Systemweit ändert sich dabei die Belastung für die Komponenten nicht. Nur für den nächsthöheren lokalen Koordinator kommt eine geringe zusätzliche Belastung hinzu. Da im Regelfall aggregierte Werte vermittelt werden sollen, entspricht die zusätzliche Belastung in etwa der Belastung eines normalen zusätzlichen IoT-Geräts. Folglich wächst das System organisch an den Stellen, an denen ein Ausbau erforderlich ist.

Da Rechen- und Speicheraufgaben dezentral ausgeführt werden, sind nur recht wenige zentrale Komponenten erforderlich. Unter der Annahme, dass mit jedem lokalen Koordinator 100 Geräte bzw. lokale Koordinatoren zusammengeführt werden könnten, ergäben sich bei nur drei Ebenen bereits eine Million miteinander verbundene Geräte, wie bereits in Unterabschnitt 6.2.2 verdeutlicht wurde. Auch in der Praxis am Gesamtdemonstrator zeigte sich die Skalierungsfähigkeit: Der lokale Koordinator der TU Braunschweig benötigt kaum Ressourcen, da er nur wenige Datenressourcen verwaltet. Auch alle anderen Koordinatoren verarbeiten jeweils nur die erforderlichen Daten.

### **Datenschutz und Datensicherheit**

Die **Anforderung NF.2** formuliert das Ziel, Anforderungen an den Datenschutz und die Datensicherheit von Nutzerinnen und Nutzern in verteilten Netzwerkstrukturen beachten zu können. Wie bereits in der Einführung zu Abschnitt 6.2 erläutert, wurde dazu die hierarchische Kommunikationsstruktur ausgearbeitet.

Die IoT-Softwareplattform entspricht der Umsetzung der lokalen Koordinatoren in der hierarchischen Kommunikationsstruktur. Durch die örtliche Begrenzung der Kommunikation mittels Regelsätzen (siehe Diskussion zu Anforderung F.3 in Unterabschnitt 6.2.3) sowie der lokalen Ausführung der Anwendungen wird es ermöglicht, sensible Daten lokal in der IoT-Softwareplattform zu verarbeiten und in ihrer Vermittlung einzuschränken. Werden Verarbeitungsinstanzen auf Basis der IoT-Softwareplattform im Sinne der hierarchischen Kommunikationsstruktur in allen lokalen Umgebungen eingebracht, können sensible Daten in diesen Umgebungen direkt verarbeitet werden und der Zugriff für Externe eingeschränkt werden.

Im Gesamtdemonstrator mussten konkrete Anforderungen an den Datenschutz und die Datensicherheit beachtet werden, die z. B. dem Schutz von Beschäftigten (§ 26 BDSG), dem Schutz der Privatsphäre (§ 22 BDSG) oder dem Schutz von Betriebsgeheimnissen (§ 1 GeschGehG) zuzuordnen sind. Die gewünschten Funktionalitäten konnten realisiert werden. Damit wird die Anforderung NF.2 als erfüllt angesehen.

### **Ausfallsicherheit**

In **Anforderung NF.3** wird eine hohe Ausfallsicherheit gefordert. Aufgrund der dezentralen Verteilung der Geräte bei der Verwendung der hierarchischen Kommunikationsstruktur können diese Geräte tendenziell schlechter gegen Hardware-Ausfälle gesichert werden, als bei einem zentralisierten System. Hier bietet ein zentralisiertes System mit einer geringeren Anzahl an Komponenten durchaus Vorteile (siehe auch Unterabschnitt 2.3.2). Zwar können auch dezentrale Komponenten durch Redundanzen gegen Ausfall gesichert werden, in der Regel ist dies jedoch nicht sinnvoll, da die zugehörigen Infrastrukturen, wie beispielsweise die Internetverbindung und die lokalen Netzwerkinfrastrukturen normalerweise auch keine Redundanzen aufweisen. Betont sei, dass auch Cloud-basierte und andere Systeme von diesen lokalen Infrastrukturen abhängig sind, wenn eine Interaktion mit lokalen Geräten stattfinden soll.

Durch die hierarchische Kommunikationsstruktur weist das System jedoch auch Vorteile hinsichtlich der Ausfallsicherheit auf. So sind bei einem Ausfall einer Teilkomponente die verbleibenden unter- und übergeordneten Systemkomponenten in der Regel weiter-

hin handlungsfähig. Für diese Fälle können die verbleibenden Anwendungen bei Bedarf Programmroutinen bzw. Szenarien vorhalten.

Zudem ist ein vollständiger Ausfall des Systems – der bei zentralen Strukturen stets berücksichtigt werden muss – durch den verteilten Charakter eher unwahrscheinlich.

Mit diesen Argumenten wird Anforderung NF.3 als erfüllt angesehen.

### **Fehlertoleranz und Robustheit**

Die Fehlertoleranz ist ein wichtiges Kriterium für den ordnungsgemäßen Betrieb komplexer Systeme. Dabei geht es darum, in geeigneter Weise auf Fehler, wie z. B. fehlende Ressourcen, zu reagieren, ohne dass der Betrieb eingestellt werden muss. In der IoT-Softwareplattform wurden dazu einige Mechanismen implementiert, die eine hohe Toleranz gegenüber Fehlern sicherstellen. Wird beispielsweise eine bestehende Netzwerkverbindung zu einer anderen Instanz getrennt, wird alle fünf Sekunden ein Versuch unternommen, die getrennte Verbindung erneut herzustellen. Dabei werden alle Datenressourcen, die über die Verbindung ausgetauscht wurden, erneut über die Verbindung bekannt gegeben. Auch Anwendungen können bei kritischen Fehlern automatisch eine Neuinitialisierung ihres Quellcodes auslösen. Damit kann auf einfache und fehlertolerante Weise auf den Wegfall von Ressourcen (bspw. in Form von Geräten, Hardware-Schnittstellen, Kommunikationspartnern) sowie sonstigen vorübergehenden Hard- und Softwarefehlern reagiert werden.

Da Anwendungen innerhalb einer Instanz der IoT-Softwareplattform gestartet werden, könnten kritische Fehler auch Auswirkungen auf andere Anwendungen haben. Deshalb wurde auf eine zuverlässige Fehlerbehandlung zwischen den Anwendungen und den sonstigen Komponenten geachtet.

Auch im Langzeitbetrieb des Gesamtdemonstrators „Real-Life-Lab Campus 2020“ waren keine Fehler zu verzeichnen, die den Betrieb grundsätzlich einschränkten. Zusammenfassend wird auch **Anforderung NF.4** zur Fehlertoleranz durch die getroffenen Maßnahmen als erfüllt angesehen.

### **Erweiterungsfähigkeit**

Die Funktionalität der IoT-Softwareplattform kann durch zusätzliche Anwendungen erweitert werden. Eine solche Erweiterungsfähigkeit durch Dritte wird in **Anforderung NF.5** gefordert. Wie bereits im vorherigen Unterabschnitt 6.2.3 erläutert, umfasst die Erweiterungsfähigkeit sowohl Techniken zur Geräteeinbindung, aber auch Funktionalitäten zum Energiemanagement. Die IoT-Softwareplattform selbst umfasst nur die notwendigsten Komponenten, wie zum Ausführen von Anwendungen (siehe Unterabschnitt 5.1.3), zur Kommunikation mit anderen Instanzen (siehe Unterabschnitt 5.1.4) sowie für die Verwaltung

von und das Zugriffsmanagement auf Datenressourcen (siehe Unterabschnitt 5.1.2). Auch diese Basiskomponenten der IoT-Softwareplattform können in ihrem Funktionsumfang durch Anwendungen erweitert werden.

Bedingt durch die einheitlichen Schnittstellen der benannten Komponenten und deren offengelegter Dokumentation kann das System auch durch Dritte mit eigenen Anwendungen erweitert werden. Diese Möglichkeit wurde beispielsweise von Projektpartnern sowie Studierenden für ihre Forschungsprojekte bzw. Abschlussarbeiten genutzt. Auch die Techniken zur Geräteeinbindung sowie die Funktionalitäten des Energiemanagements wurden in Form solcher Anwendungen umgesetzt (siehe Kapitel 5).

### **Implementierungsaufwand**

Mit **Anforderung NF.6** wird ein möglichst geringer Implementierungsaufwand für die Realisierung der Anwendungen gefordert. Nachfolgend wird dargelegt, welche Maßnahmen unternommen wurden, um den Implementierungsaufwand neuer Anwendungen gering zu halten.

Die IoT-Softwareplattform wurde auf JavaScript und anderen Webtechnologien aufgebaut. Auch die Anwendungen werden mittels dieser Sprachen und Techniken umgesetzt. Diese Sprachen gelten nach [Sch+15, S. 18] im Vergleich zu nativen Anwendungssprachen als „leichtgewichtig“ und erfordern „lediglich einen geringen Implementierungsaufwand“. Auch wird in der zitierten Quelle benannt, dass das Erlernen dieser Sprache leichter als bei nativen Anwendungssprachen ist.

Die Definition der Schnittstellen zwischen den Komponenten der IoT-Softwareplattform und den Anwendungen (siehe Abschnitt 5.1) wurde möglichst einfach gehalten, damit der Umgang mit diesen schnell zu erlernen ist. Wie aus den Beschreibungen der Schnittstellen in Unterabschnitt 5.1.2 und 5.1.3 zu entnehmen ist, werden nur wenige universelle Basismethoden verwendet. Die vorhandene Dokumentation dieser notwendigen Schnittstellen und die Bereitstellung von Quellcode-Beispielen erleichtern dabei zusätzlich die Entwicklung.

Weiterhin sind Anwendungen in der Lage, Funktionen voneinander zu erben. Hierzu wurden einige Basis-Anwendungen erstellt, die wiederkehrende Strukturen und Funktionen zur Verfügung stellen.

### **Installierbarkeit und Aktualisierung**

Alle Anwendungen sowie auch die IoT-Softwareplattform selbst wurden als Pakete des Node.js-Paketmanagers NPM (siehe Unterabschnitt 5.1.3) entwickelt. Die IoT-Softwareplattform, die Anwendungen und alle Abhängigkeiten lassen sich damit mit nur wenigen Befehlen installieren. Anwendungen, wie [install-apps-git](#) und [install-apps-npm](#) (siehe

auch Unterabschnitt 5.1.7) ermöglichen zudem die automatische Installation von benötigten Anwendungen. Über den Paketmanager NPM oder über die Versionsverwaltung GIT lassen sich zudem alle Komponenten aktualisieren. Diese Möglichkeiten der Installierbarkeit und Aktualisierung werden in **Anforderung NF.7** gefordert.

Die IoT-Softwareplattform kann auf allen gängigen Betriebssystemen ausgeführt werden (siehe die Einführung zu Kapitel 5). Sie ist damit sowohl auf Desktop-Computern, Servern und auch auf Mini-Computern, wie dem Raspberry Pi lauffähig. Mittels des Frameworks zur Darstellung von Nutzerschnittstellen können die Funktionen der IoT-Softwareplattform auch in Webbrowsern, auf Smartphones und Tablets ausgeführt werden. Auf diese Weise konnten

- erste intelligente (Haushalts-)Geräte,
- Vermittler zu anderen Bussystemen und Infrastrukturen, sogenannten Bridges (engl. für Brücke),
- Server-Anwendungen,
- Web-Anwendungen,
- mobile Smartphone- und Tablet-Anwendungen,
- Desktop-Anwendungen und
- Display-Anwendungen entwickelt werden.

Wie in Unterabschnitt 2.3.2 dargestellt, weisen zentralisierte Strukturen Vorteile hinsichtlich einer einfachen Software-Bereitstellung für die Plattformbetreiber auf. In einer verteilten Struktur ist die Bereitstellung von Anwendungen naturgemäß deutlich aufwendiger. Wie bei der Diskussion der Anforderungen zur Erweiterungsfähigkeit, zum Implementierungsaufwand sowie zur Installierbarkeit und Aktualisierung verdeutlicht, wurde ein großer Aufwand betrieben, um diesen Nachteil möglichst gering zu halten. Dennoch lässt sich ein zusätzlicher Aufwand nicht gänzlich vermeiden.

## 6.2.5 Fazit

Die hierarchische Kommunikationsstruktur ist ein geeignetes Konzept für die Kommunikation und Verarbeitung von verteilten Daten. Anforderungen an den Datenschutz und die Datensicherheit können feingranular modelliert und beachtet werden. Wie beabsichtigt können damit Daten in ihren lokalen Umgebungen verarbeitet und gehalten werden. Die Nutzerinnen und Nutzer behalten dabei die Kontrolle über ihre eigenen Daten.

Neben der Zielsetzung, Anforderungen zum Datenschutz und zur Datensicherheit beachten zu können, weist die hierarchische Kommunikationsstruktur weitere Vorteile auf. So ist aufgrund der vielfältigeren Vernetzungsmöglichkeiten im Vergleich zu den in Unterabschnitt 2.3.5 definierten Systemtypen (siehe Unterabschnitt 6.2.2) die Skalierungsfähigkeit erheblich höher. Das System kann so dynamischer eingesetzt und in einer größeren Anzahl von Umgebungen implementiert werden. Durch den verteilten Charakter sind die Gesamtstruktur und die Teilstrukturen im Vergleich zu den anderen Systemtypen aus Unterabschnitt 2.3.5 zudem besser gegen Ausfall geschützt.

Durch die Möglichkeit, die Struktur feingradig zu unterteilen, können zudem Latenzanforderungen und Anforderungen an Netzwerkkapazitäten individuell adressiert werden. So könnten insbesondere in lokalen Netzen sogar weiche Echtzeitanforderungen<sup>4</sup> mit geringen Latenzen realisiert werden.

Die IoT-Softwareplattform erfüllt die an sie gestellten Anforderungen einwandfrei (siehe Unterabschnitt 6.2.3). Auch die nicht-funktionalen Anforderungen aus Unterabschnitt 3.3.3 wurden erfüllt (siehe Unterabschnitt 6.2.4). Die Diskussionen zur Umsetzung haben darüber hinaus gezeigt, dass viele Anforderungen wesentlich umfänglicher erfüllt wurden, als dies für ein Test-Framework zur Erprobung des Energiemanagements notwendig gewesen wäre. Dies gilt insbesondere für die Fehlertoleranz, den Implementierungsaufwand, die Installierbarkeit und Aktualisierung sowie für die Latenzanforderungen. Dadurch konnten Nachteile in der Software-Bereitstellung aufgrund der verteilten Struktur ausgeglichen werden. Durch die Kapselung der Funktionen, der transparenten Kommunikation und der Begrenzung der Kommunikation wurden nicht nur die Grundmerkmale der hierarchischen Kommunikationsstruktur realisiert, sondern auch eine vielseitig einsetzbare Softwareplattform für eine große Zahl verschiedenster Anwendungsfälle geschaffen. Funktionen für das Auto Discovery ermöglichen zudem eine systemweite einheitliche Möglichkeit zur automatischen Verknüpfung von Komponenten und damit eine Konfiguration mit einem möglichst geringeren Aufwand.

Durch die wesentlich umfängliche Erfüllung der Anforderungen eignet sich die IoT-Softwareplattform auch für den realen Einsatz. Dies wird auch durch die hohe Leistungsfähigkeit bei der Paketvermittlung bestätigt (siehe Unterabschnitt 6.2.1). Durch geringe Latenzen, durch geringe Werte des Jitters sowie durch eine hohe Zahl an verarbeiteten Nachrichten pro Sekunde kann die Plattform im Vergleich zu weitverbreiteten quelloffenen Softwareprodukten zur Datenvermittlung bestehen.

---

<sup>4</sup>Bei einem Echtzeitsystem muss eine Aufgabe in einer vorgegeben Zeit abgearbeitet werden, um eine Funktionalität zu gewährleisten (siehe auch Unterabschnitt 3.3.4). Bei weichen Echtzeitsystemen ist dieser Zwang nicht gegeben. So soll die vorgegebene Verarbeitungszeit zwar im Normalfall eingehalten werden, das System arbeitet aber weiterhin einwandfrei, wenn die Verarbeitungszeit in Einzelfällen überschritten wird (vgl. [Sch05, S. 40]).

Die unbeschränkte Erweiterungsfähigkeit der IoT-Softwareplattform in Form von Anwendungen ist eine weitere wertvolle Eigenschaft, da dadurch der mögliche Funktionsumfang kaum beschränkt ist. Es können Anwendungen aller Schichten und sogar weit jenseits des Energiemanagement realisiert werden. Mögliche Erweiterungen der Plattform können jederzeit auch von Dritten durchgeführt werden. Anwendungen können dabei auch Ansichten enthalten, mit denen Nutzerinnen und Nutzern graphische Informationen bereitgestellt werden können. Zur Anzeige wird ein eigenes Framework auf Basis der IoT-Softwareplattform verwendet. Somit ist es möglich, eine Funktionalität inkl. der Nutzeroberfläche vollständig auf Basis der IoT-Softwareplattform zu entwickeln.

Mit diesen hervorzuhebenden Konzepten und Umsetzungen können Funktionalitäten realisiert werden, die verschiedene Anwendungsbereiche, mehrere Umgebungen sowie mehrere Gerätetypen und -hersteller umfassen. Die Marktbeschränkungen bisheriger Betreiber von Kommunikationsplattformen im Bereich des Smart Home und des Energiemanagements im Privathaushalt (siehe auch Kapitel 1 und Abschnitt 3.2) könnten damit in Frage gestellt und die Basis für offener Systemwelten gelegt werden. Insgesamt betrachtet, könnten damit die zukünftig erforderlichen Innovationen im Bereich des Internet der Dinge deutlich motiviert werden.

## 6.3 Untersuchung der Verfahren zur Einbindung von Geräten

Neben der Ausarbeitung der Kommunikationsstruktur fokussiert diese Arbeit auf der Einbindung von Geräten (siehe Schicht der Geräteeinbindung in Abschnitt 3.3). Unterabschnitt 3.3.2 benennt, dass prinzipiell alle Gerätearten eingebunden werden sollten und unterscheidet dazu vier verschiedene Geräteklassen anhand ihrer Kommunikationsanbindung und ihres Kommunikationsbedarfs. Diese unterliegen aufgrund ihrer Kommunikationsfähigkeiten und ihres Verhaltens jeweils anderen Herausforderungen. In diesem Abschnitt soll überprüft werden, ob die verschiedenen Verfahren ihre Funktionstüchtigkeit bestätigen können. Durch den notwendigen Funktionsumfang aus Unterabschnitt 4.1.1 wird deutlich, welche Daten oder Funktionen bei einer Einbindung fehlen und welche Einbindungsverfahren verwendet werden können.

Als **intelligente Geräte** werden in Unterabschnitt 3.3.2 Geräte definiert, deren Kommunikationsanbindung vollständig kompatibel mit dem notwendigen Funktionsumfang ist. Existente Geräte, die diese Anforderung gänzlich erfüllen, wurden im Rahmen dieser Arbeit nicht



identifiziert. Mit dem umgerüsteten Kühltank (siehe Unterabschnitt 6.1.2) wurde jedoch ein eigenes erstes intelligentes Gerät auf Basis der IoT-Softwareplattform geschaffen.

**Vernetzte Geräte** können direkt oder über vorhandene Kommunikations- und Bussysteme eingebunden werden. In einigen Fällen können die vorhandenen Kommunikationssysteme automatisch erkennen, welche Geräte sich in ihrem Netzsegment befinden. In den meisten Fällen müsste die Nutzerin bzw. der Nutzer die Geräte jedoch selbst adressieren und auch passende Gerätemodelle hinterlegen bzw. auswählen. Über die implementierten Bussysteme (siehe Unterabschnitt 5.1.7), wie EtherCAT, Modbus, Midi und I2C können die Betriebszustände und ggfs. weitere Daten sowie Steuerungsbefehle übertragen werden. Jedoch kann nicht jedes Gerät alle diese Zugriffe ermöglichen. Mitunter ist z. B. nur ein Steuerungszugriff möglich. Über die Bussysteme wie KNX, Midi, DMX und ArtNet können ebenfalls nur Betriebszustände vorgegeben werden. Damit sind diese dem System jedoch bekannt, solange das Bussystem bzw. Geräte keine Steuerungen anderer Parteien zulassen. Steht die Information über den Energieverbrauch eines Gerätes nicht zur Verfügung und soll kein zusätzliches Energiemessgerät verbaut werden, können die Techniken zur Ermittlung des Energieverbrauchs aus Abschnitt 5.2 angewendet werden. Deren Untersuchung wird nachfolgend dargestellt.

Die Technik zur direkten Zuordnung des Energieverbrauchs aus Unterabschnitt 5.2.3 wird aufgrund ihrer Trivialität in dieser Arbeit nicht näher evaluiert. Sie ist aufgrund ihrer Definition ausschließlich auf Geräte anwendbar, deren Energieverbrauch über die Zeit bei gleichem Betriebsstatus nicht variiert. Abweichungen können sich aus den Messabweichungen bei der Kalibrierung oder aus Abweichungen zu den Angaben des Energieverbrauchs aus dem Typenschild bzw. der Betriebsanleitung ergeben. Zudem können sich Abweichungen aufgrund von thermischen Effekten sowie aufgrund von Alterung ergeben. Im Gesamtdemonstrator wurde diese Technik auf ausgewählte Lampen und Lüfter, die im Sinne des Smart Home eingebunden waren, auf bereits vernetzte IT-Geräte, wie Router und Switches, sowie auf die Geräte einer gesteuerten Multimediaanlage (siehe Medientechnisches Projekt [13/703]) angewendet.

Das Verfahren zur Energieverbrauchsermittlung von Desktop-Computern wird in Unterabschnitt 6.3.2 näher untersucht. Zur Vereinfachung der Analyse wird ausschließlich die Einflussgröße der CPU-Last variiert.

**Statische Geräte** verändern ihren Betriebsstatus nicht und weisen auch sonst kein veränderndes Verhalten auf. Dementsprechend wird auch der Energieverbrauch als statisch angenommen. Die oben benannte Zuordnung des Energieverbrauchs kann hier ebenfalls verwendet werden. Auch hier wird mit ähnlichen Messabweichungen aus den gleichen Gründen

gerechnet, weshalb auf eine detaillierte Evaluation ebenfalls verzichtet wird. Im Gesamtdemonstrator wurde diese Einbindung für permanent in Betrieb befindliche Geräte, wie ausgewählte Pumpen, Lüfter und die Notfallbeleuchtung gewählt.

Sonstige **nicht-vernetzte Geräte** sollen nach Abschnitt 4.2 mittels der sogenannten IoT-Gerätetrepräsentation nachgebildet werden. Die Nutzerin bzw. der Nutzer muss auch hier die Geräte selbst angeben und passende Modellierungen hinterlegen. Kernelement der Einbindung ist die Ermittlung des Betriebsstatus, die auch als Zustandsidentifikation bezeichnet wird. In Abschnitt 5.2 wurden mehrere solcher Techniken vorgestellt, die nachfolgend untersucht werden. In Unterabschnitt 6.3.3 wird die Zustandsidentifikation mittels Schwellenwert-Klassifikation aus Unterabschnitt 5.2.6 anhand des Modells aus Unterabschnitt 5.2.7 evaluiert. In Unterabschnitt 6.3.4 wird anschließend die Zustandsidentifikation anhand des DTW-Musterabgleichs aus Unterabschnitt 5.2.8 betrachtet. Beide Verfahren ermitteln den Betriebszustand aus zusätzlichen Energiemessgeräten. In Unterabschnitt 6.3.5 wird die Mehrgeräteerkennung mittels künstlicher neuronaler Netze anhand von Netzspannungseinflussungen aus Unterabschnitt 5.2.9 analysiert.

Aus diesen Beschreibungen wird bereits deutlich, dass alle in Unterabschnitt 3.3.2 geforderten Gerätearten abgedeckt werden. Durch die Kombination der unterschiedlichen Verfahren erscheint es möglich verschiedenste Geräte kostengünstig einzubinden. Es ist weder eine Neuanschaffung aller Geräte, noch eine Ausrüstung aller Geräte mit einzelnen Energiemesssteckdosen erforderlich.

Da einige Einbindungen auf Schätzverfahren basieren, soll bei den Untersuchungen in den folgenden Unterabschnitten insbesondere die Qualität der Schätzungen untersucht werden. Diese Untersuchungen sollen anhand verschiedener kleiner Testumgebungen oder anhand von geeigneten Datensets vorgenommen werden. Unter Datensets werden Datenbanken mit Aufnahmen der Energieverbräuche verschiedener Geräte verstanden. Die verwendeten Hilfsanwendungen zur Nutzung der Datensets werden in Unterabschnitt 6.3.1 vorgestellt. In Unterabschnitt 6.3.6 wird abschließend ein Fazit gegeben.

### 6.3.1 Evaluationsumgebung & Hilfsanwendungen

Die überwiegende Anzahl der in dieser Arbeit implementierten Erkennungs- und Nachbildungsverfahren wurden für die direkte Verarbeitung der Daten ausgelegt und optimiert (siehe Unterabschnitt 5.2.1). Für die Evaluation mit zuvor aufgenommenen Messdaten und Wahrheiten mussten deshalb zusätzliche Strukturen geschaffen werden, die die Messdaten als Messereignisse in die Algorithmen speisen.

Mit den Hilfsanwendungen **json-save** sowie **csv-save** können alle Daten, die in einer Datenressource veröffentlicht werden, in einer Datei im JSON- oder CSV-Format gespeichert werden. Die Hilfsanwendung **json-play** erlaubt die spätere erneute Veröffentlichung aller Daten einer solchen JSON-Datei. Mittels der Hilfsanwendung **save-tree-as-object** können die Daten mehrerer Datenressourcen in einer Datenressource zusammengefasst werden.

### 6.3.2 Energieverbrauchsermittlung von Desktop-Computern

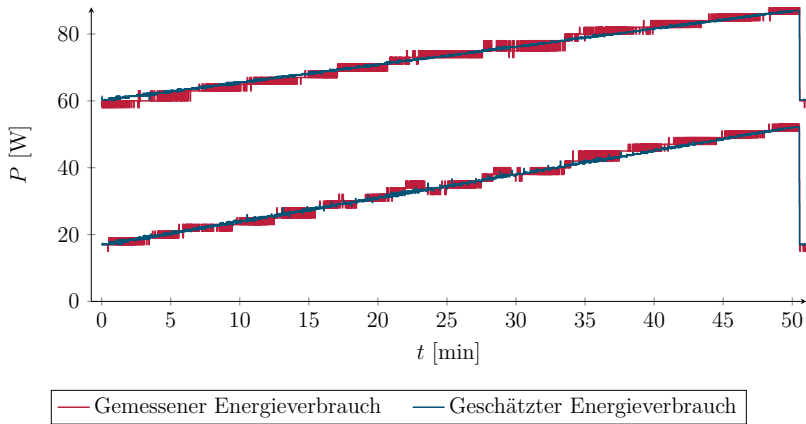


Abbildung 6.7: Darstellung des Messzyklus ansteigend von 0 bis 100 % CPU-Last und direkt zurück auf 0 % CPU-Last in sekundlicher Auflösung der Computer A-1 (oben) und D-1 (unten).

Nachfolgend wird die Energieverbrauchsermittlung von Desktop-Computern aus Unterabschnitt 5.2.4 betrachtet. Dazu wurde die Umsetzung auf 17 Computern aus fünf verschiedenen Produktserien namhafter Markenhersteller angewendet. Mittels des nachfolgend beschriebenen Hilfsanwendung wurde dabei ein Messzyklus von 0 bis 100 % CPU-Last durchlaufen, wie in Abbildung 6.7 beispielhaft visualisiert. Auf die Variation der GPU-Last wurde verzichtet, da sie mit erheblich höherem Aufwand verbunden ist und die Aussagekraft der Analyse nicht weiter verbessert, wie eine Voruntersuchung gezeigt hat.

Mit der Hilfsanwendung **pc-energy-estimation/stress-cpu** kann automatisch eine gewünschte CPU-Last über alle Prozessorkerne hinweg auf dem zu testenden Computer erzeugt werden. Damit die Hilfsanwendung korrekt arbeitet, musste auf den Computern der zwei modernsten Produktserien die dynamische Frequenzwahl der CPU deaktiviert werden. Dies führte jedoch nicht zu einer Änderung der Leistungsaufnahme. Da das Hilfsprogramm als Anwendung der IoT-Softwareplattform umgesetzt wurde, kann die CPU-Last auch aus der

Ferne variiert werden. Auch die Energieverbrauchsmessung und Evaluation wurden auf Basis der IoT-Softwareplattform durchgeführt.

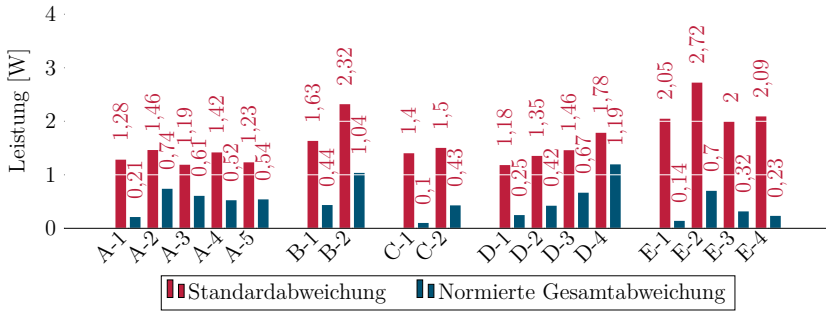


Abbildung 6.8: Abweichung von geschätztem zu tatsächlichem Energieverbrauch pro Computer. Die Standardabweichung in Rot wurde über die Einzelwerte ermittelt. Die normierte Gesamtabweichung in Blau wurde über die gesamte Datenreihe akkumuliert. Für die Ermittlung wurde eine lineare Schätzung verwendet.

In Abbildung 6.8 sind die Standardabweichung der Einzelwerte sowie die normierte Gesamtabweichung des geschätzten zum tatsächlichen Energieverbrauchs für jeden untersuchten Computer aufgetragen. Dabei wurden jeweils mehrere Computer einer Modellreihe untersucht. Für die Schätzung wurde zunächst eine lineare Schätzung verwendet. Die Schätzung wurde jeweils mit einer initialen Messung bei 0 und 100 % CPU-Last kalibriert. Die hohe Standardabweichung der Einzelwerte (in rot) ergibt sich durch die geringe Messauflösung der Einzelwerte der verwendeten Messgeräte von zwei bzw. drei Watt. Diese geringere Auflösung ist auch in Abbildung 6.7 erkennbar.

Normiert auf die mittlere Leistungsaufnahme ergibt sich eine Gesamtabweichung von 1,04 % über alle Computer. Zu beachten ist, dass für die Kalibrierung und für die Erprobung das gleiche Hilfsprogramm zur Auslastung der CPU verwendet wurde. Bei anderen Rechenaufgaben ist insbesondere bei modernen CPUs mit Abweichungen zu rechnen. Dennoch zeigt diese Evaluation auf, dass das zugrundeliegende Konzept sinnvoll ist und dieses Nachbildungsverfahren eingesetzt werden sollte.

### 6.3.3 Zustandsidentifikation mittels Schwellenwert-Klassifikation anhand des Modells aus Unterabschnitt 5.2.7

Nachfolgend wird die Zustandsidentifikation mittels Schwellenwert-Klassifikation aus Unterabschnitt 5.2.6 auf Basis von Energiemessgeräten betrachtet. Dabei wird das Modell aus

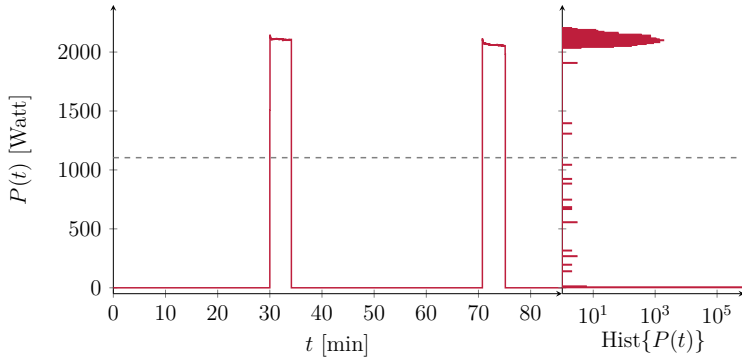


Abbildung 6.9: Ausschnitt des zeitlichen Verlaufs der Leistungsaufnahme eines Wasserkochers mit zwei Aktivitäten (links) sowie Histogramm der Leistungsaufnahme über 17 Tage (rechts). In Grau ist eine exemplarische Entscheidungsschwelle bei 1.100 Watt eingezeichnet.

Unterabschnitt 5.2.7 angewendet. Für die Evaluation wird die Tracebase-Datenbank aus [Rei+12] bestehend aus Messdaten für 157 verschiedene Geräte verwendet.<sup>5</sup> Die Messdaten wurden mit Abtastraten zwischen einer und fünf Sekunden aufgezeichnet. Die Messdauern variieren zwischen wenigen Tagen bis zu einem Monat.

Die Abbildung 6.9 zeigt die Leistungsaufnahme eines Wasserkochers in ihrem zeitlichen Verlauf sowie als Histogramm dargestellt. Unter Nutzung der Zustandsidentifikation mittels Schwellenwert-Klassifikation aus Unterabschnitt 5.2.6 kann z. B. die in Grau eingezeichnete Entscheidungsschwelle bei 1.100 Watt verwendet werden. Wie im Histogramm ersichtlich, zeigt die Leistungsaufnahme zwei Häufungspunkte, die auf die zwei Ausprägungen Ein und Aus des Betriebsstatus zurückzuführen sind. Demzufolge lässt sich der Betriebsstatus des Gerätes mittels dieser Technik anhand der dargestellten Entscheidungsschwelle zuordnen. Im abgebildeten Ausschnitt würden demnach zwei Aktivitäten detektiert.

Auf Basis des Modells aus Unterabschnitt 5.2.7 konnte auf ähnliche Weise für viele weitere Geräte der jeweils momentane Betriebsstatus über die Zeit zugeordnet werden. Von den 157 Geräten des Tracebase-Datensatzes konnten 115 Geräte, also 73 % des gesamten Datensatzes, vollständig zugeordnet werden. Die Kontrolle der Erkennung wurde manuell im Abgleich mit dem jeweiligen Energieverbrauch vorgenommen. Bei der vorgenommenen Aufteilung in Aufgabenbereiche in Abbildung 6.10 fällt auf, dass insbesondere die Kategorien Kühlen, Waschen und die Kaffeeautomaten schlechte Erkennungsraten aufweisen. Diese Kategorien werden dadurch charakterisiert, dass die Geräte einem internen Programmablauf oder wie bei

<sup>5</sup>Zu einem Gerät waren keine sinnvollen Messdaten hinterlegt.

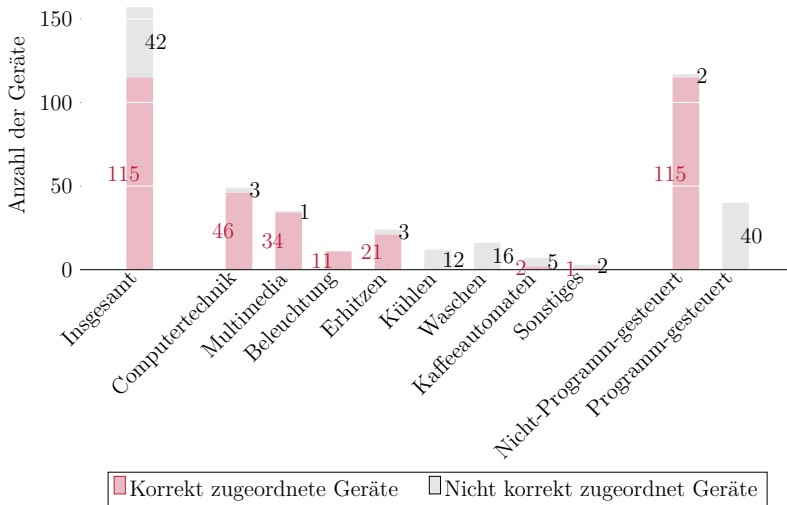


Abbildung 6.10: Darstellung der Anzahlen der Geräten, deren jeweils momentaner Betriebsstatus vollständig korrekt erkannt werden konnte, im Vergleich zu den verbleibenden Geräte unter Verwendung des Modells aus Unterabschnitt 5.2.7 aufgeteilt nach verschiedenen Kategorien: Links – alle Geräte, Mitte – Aufteilung der Geräte nach Aufgaben, Rechts – Aufteilung in Nicht-Programm-gesteuerte und Programm-gesteuerte Geräte.

den Kühlschränken einer internen Regelung unterliegen. In Abbildung 6.10 wird zusätzlich in Programm- und Nicht-Programm-gesteuerte Geräte unterschieden. Dies bestätigt die Vermutung, dass sich Programm-gesteuerte Geräte nicht durch das Modell aus Unterabschnitt 5.2.7 erkennen lassen. Verständlich wird dieser Sachverhalt an der Abbildung 6.11, die exemplarisch die Leistungsaufnahme einer Waschmaschine dargestellt. Dabei wird deutlich, dass die eingezeichnete Entscheidungsschwelle den Betriebsstatus nicht korrekt abbildet. Dies ist unter anderem darauf zurückzuführen, dass das Waschprogramm Ruhephasen ohne Leistungsaufnahme aufweist und damit kein Unterschied vom ausgeschalteten Betriebsstatus festzustellen ist.

Betrachtet man folglich ausschließlich die nicht-programmgesteuerten Geräte, werden nur lediglich zwei Geräte durch diese Zustandsidentifikation nicht vollständig korrekt erkannt, was einer Erkennungsrate von 98,3 % entspricht.

Mit angepassten Modellen könnten weiterhin auch Programm-gesteuerte Geräte sowie permanent in Betrieb befindliche Geräte erkannt werden. Geräte, die dauerhaft laufen und einer internen Regelung unterliegen, könnten durch das Ausbleiben einer Aktivität zugeordnet werden.

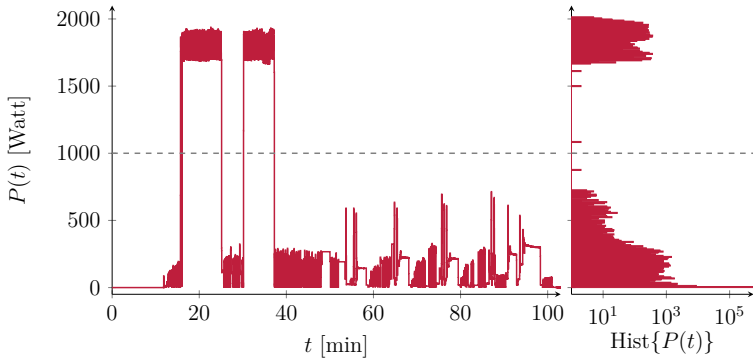


Abbildung 6.11: Ausschnitt des zeitlichen Verlaufs der Leistungsaufnahme einer Waschmaschine (links). Dargestellt ist ein einzelnes Waschprogramm. Zusätzlich ist die Leistungsaufnahme über 10 Tage als Histogramm abgebildet (rechts). Die in Grau dargestellte exemplarische Entscheidungsschwelle bei 1.000 Watt bildet den Betriebsstatus nicht korrekt ab.

### 6.3.4 Zustandsidentifikation anhand des DTW-Musterabgleichs

In diesem Unterabschnitt soll die Zustandsidentifikation anhand des DTW-Musterabgleiches (siehe Unterabschnitt 5.2.8) erprobt werden, mit der eine bessere Erfassung des momentanen Zustands der Programm-gesteuerten Geräte angestrebt wird. Für die Evaluation wird der Energieverbrauch vieler Programmabläufe eines Gerätes und mit einer Kennzeichnung der tatsächlichen gewählten Programme benötigt. Dazu wurden Messdaten einer Waschmaschine eines Drei-Personen-Haushalts über einen Zeitraum von acht Monaten ausgewertet. Die Nutzerinnen und Nutzer waren dazu aufgefordert worden, die gewählten Waschprogramme und die sonstigen Einstellungen zu protokollieren.

Durch Störungen des verwendeten funkbasierten Energiemesssystems kam es zeitweise zu fehlenden Messwerten. Ausgeschlossen wurden deshalb Messreihen, in denen mehr als 30 % der Messwerte fehlten. Bei den restlichen Messreihen wurden fehlende Datenpunkte durch Wiederholungen des vorangegangenen Wertes kompensiert.

Durch den DTW-Musterabgleich konnte eine korrekte Zuordnung von 89,1 % mit einer Unsicherheit von 4,3 Prozentpunkten der 76 zu erkennenden Waschgänge erreicht werden. Dabei wurden sieben verschiedene Waschprogramme unterschieden. Die Ermittlung wurde über 500 zufalls-initiierte Erkennungsdurchläufe gemittelt. Um eine Separation in unabhängige Trainings- und Testdaten zu erhalten, wurden zufällig Messreihen eines Waschprogramms als Muster festgelegt und vor der durchzuführenden Erkennung aus dem Datenpool der Testdaten entfernt.

Zusätzlich wurde überprüft, wie viele weitere Muster benötigt werden, um alle Waschgänge zuzuordnen. Zu beachten ist, dass auch hier jedes zusätzliche Muster aus dem Datenpool der Testdaten entfernt werden muss. Dabei verringert sich die oben genannte Anzahl von 76 zu erkennenden Waschgängen. Mit 2,4 zusätzlichen Mustern bei einer Unsicherheit von 1,3 Mustern konnten auch alle verbleibenden Waschgänge (im Durchschnitt 8,8 Waschgänge:  $103 \cdot (1 - 0,891) - 2,4 = 8,8$ ) erkannt werden. Auch hier wurde die Ermittlung über 500 zufalls-initiierte Erkennungsdurchläufe durchgeführt. Es kann festgestellt werden, dass der verwendete DTW-Algorithmus sehr wenige Muster für die Erkennung benötigt.

Der DTW-Musterabgleich zeichnet sich dadurch aus, dass die Längen der Sequenzen angepasst werden. Diese Anpassung hat den Vorteil kurzzeitige bzw. leichte zeitliche Abweichungen im Programmablauf auszugleichen. Bei Betrachtung der Sequenzen wird tatsächlich deutlich, dass die Sequenzen eines identischen Waschprogramms variieren. Dieses könnte an unterschiedlichen Füllmengen oder auch an unterschiedlichen Verschmutzungsgeraden der Wäsche liegen, auf die der Programmablauf adaptiv reagiert. Die Wahl des DTW-Musterabgleiches ist im Vergleich zu anderen Verfahren, wie der Kreuzkorrelation also sinnvoll.

### 6.3.5 Mehrgeräteerkennung mittels künstlicher neuronaler Netze anhand von Netzspannungsbeeinflussungen

In Unterabschnitt 5.2.9 wurde die Mehrgeräteerkennung anhand von elektromagnetischen Netzspannungsbeeinflussungen vorgestellt. Die hier verwendete Messkampagne wurde gemeinsam mit einem Studenten in der Masterarbeit [16/015] durchgeführt. Für die Evaluation wurde der in Unterabschnitt 5.2.9 beschriebene Mess- und Demonstrationsaufbau verwendet. Mit Hilfe des Aufbaus wurden verschiedene Geräte vermessen. Dabei wurde festgestellt, dass es zwei Gerätekategorien gibt, die sich nicht für die Erkennung eignen:

- Geräte, die keine oder zu geringe elektromagnetische Beeinflussungen aufweisen und deshalb nicht erkannt werden können. Diese können in der Regel auch weiterhin im Messaufbau betrieben werden, ohne die Erkennung anderer Geräte zu stören,
- Geräte, die zu hohe elektromagnetische Beeinflussungen emittieren und damit die Erkennung anderer Geräte unmöglich machen. Im Test wurde dieser Kategorie eine LED-Lampe zugeordnet, die den Radio-Empfang durch hohe breitbandige Aussendungen störte.

Bei den verbliebenen nicht zugeordneten Geräten handelte es sich überwiegend um elektronische Geräte. Viele elektronische Geräte besitzen interne getaktete Spannungswandler.



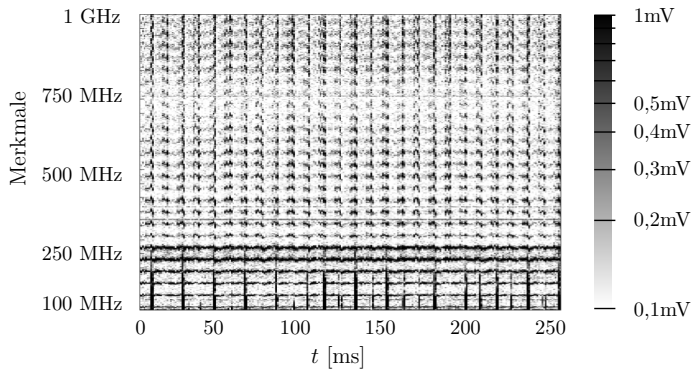


Abbildung 6.12: Frequenz-basierte Merkmale eines USB-Netzteils im Verlauf über 0,25 Sekunden. Die Darstellung der Helligkeitsskala wurde logarithmiert. Die Skala wurde auf den Eingangspegel zurückgerechnet.

Durch ihre periodische Regelung der Ausgangsspannung erzeugen sie hochfrequente Signalcomponenten, die in das Niederspannungsnetz eingekoppelt werden. In Abbildung 6.12 sind exemplarisch die elektromagnetischen Beeinflussungen eines USB-Netzteils mit Lastwiderständen als Frequenz-basierte Merkmale über eine Zeitspanne von 0,25 Sekunden dargestellt. Aus der Abbildung wird deutlich, dass sich periodisch wiederholende Charakteristiken ausprägen, anhand welcher eine Identifikation unterschiedlicher Geräte vorgenommen werden kann.

Für den Demonstrationsaufbau wurden verwendet:

- Ein PC-Monitor 24 Zoll,
- Eine externe Festplatte mit Netzteil,
- Ein USB-Netzteil mit Lastwiderständen,
- Ein USB-Hub mit Netzteil und
- Eine LED-Lampe mit E14-Fassung.

Zu beachten ist, dass die im Demonstrationsaufbau verwendete Anzahl der Geräte nicht groß genug ist, um repräsentative Ergebnisse zu erzeugen. Jedoch zeigt die qualitative Analyse der Erkennungsergebnisse einige Vor- und Nachteile dieser Technik, die im weiteren Verlauf dargelegt werden.

Alle Geräte wurden für das Training einzeln vermessen. In den ersten Untersuchungen war festzustellen, dass sich aufgrund thermischer Effekte die Netzspannungsbeeinflussungen nach

dem Einschalten teilweise noch verändern. Deshalb wurden die Messdaten über mehrere Stunden aufgenommen und anschließend konsolidiert. Wie in Unterabschnitt 5.2.9 benannt, wurden die Messdaten rechnerisch in allen Kombinationen miteinander addiert und ein einzelnes künstliches neuronales Netz berechnet.

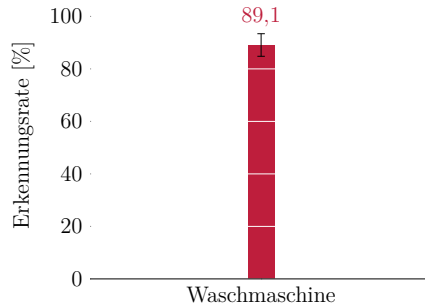


Abbildung 6.13: Darstellung der Fehlerrate bei der Zuordnung von einzelnen Geräten und verschiedene Kombinationen von Geräten sowie die Gesamtfehlerrate über alle Kombinationen.

Für die Erkennung von Gerätekombinationen wurden verschiedene Geräte gleichzeitig betrieben. Die Abbildung 6.13 zeigt verschiedene Fehlerraten bei der Zuordnung. Eine Zuordnung wurde als fehlerhaft angenommen, wenn nicht alle Gerätezustände vollständig korrekt erkannt wurden. Dargestellt wird die Zuordnung in folgenden Varianten: bei jeweils einem aktiven Gerät, bei vier aktiven Geräten, eine Zuordnung mit allen Geräten und eine Leermessung ohne aktive Geräte. Darüber hinaus wurde ein Durchschnittswert für alle Kombinationen errechnet. Bei der Nutzung von künstlichen neuronalen Netzen sind Rückschlüsse auf Basis der Ergebnisse nur bedingt möglich, da nicht direkt herausgestellt werden kann, auf welche konkreten Ereignisse in den Merkmalen das Netz optimiert wurde. Dies wird beispielsweise an der erhöhten Fehlerrate der Einzelgeräterkennung der externen Festplatte deutlich. Aus den Ergebnissen ist dennoch zu erkennen, dass einige Kombinationen bereits mit sehr geringen Fehlerraten zugeordnet wurden. Einige Kombinationen weisen hingegen teils sehr schlechte Fehlerraten auf. Gute Erkennungsraten traten auf, wenn der PC-Monitor und der USB-Hub nicht gemeinsam aktiv waren, was auf eine gegenseitige Beeinflussung der beiden Geräte hinweist.

Deshalb wurden in Abbildung 6.14 die Modellausgabewerte der Kombinationen aller Geräte mit und ohne PC-Monitor gemittelt über 512 Rahmen dargestellt. Aus der Abbildung wird bestätigt, dass die Zuordnung des USB-Netzteils abhängig von der Aktivität des PC-Monitors ist. Der Monitor enthält einen Filter zur Leistungsfaktorkorrektur (engl. Power Factor Correction, PFC). Dieser soll unter anderem die elektromagnetischen Netzspannungs-

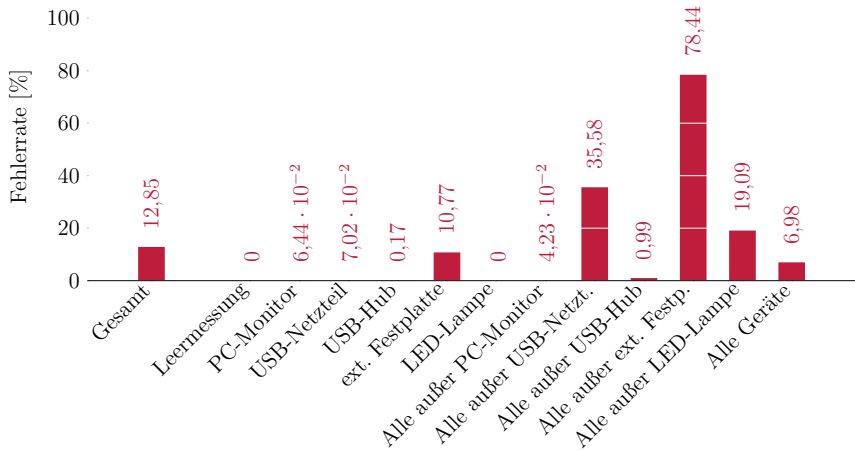


Abbildung 6.14: Darstellung der Erkennung aller Geräte gleichzeitig im Vergleich zur Erkennung aller Geräte ohne den PC-Monitor. Aufgetragen sind die Modellausgabewerte des künstlichen neuronalen Netzes der zugeordneten Geräte. Der dargestellte Wert wurde jeweils aus 512 Rahmen gemittelt. Auffällig ist die Änderung des Erkennungswerts in Abhängigkeit der Aktivität des Monitors beim hier betrachteten USB-Ladegerät. Dabei ist festzustellen, dass die Erkennung aller Geräte ohne eingesteckten PC-Monitor eine sehr gute Fehlerrate aufweist.

beeinflussungen des Monitors kompensieren. Dieser Filter wirkt jedoch auch auf andere benachbarte Geräte, wie aus der Betrachtung der Messdaten deutlich wird. So verändert der an das 230-V-Versorgungsnetz angeschlossene, aber ausgeschaltete PC-Monitor bereits die Beeinflussungen, die vom USB-Netzteil ausgesendet werden. Auch die Netzspannungsbeeinflussungen anderer Geräte werden gedämpft, wodurch die Vorbedingung der Unabhängigkeit der Geräte nicht gegeben ist. Aus dieser Erkenntnis wird aus Abbildung 6.13 deutlich, dass unter Vernachlässigung der Kombinationen mit aktivem PC-Monitor diese Technik zur Mehrgeräteerkennung prinzipiell sehr gute Ergebnisse liefert.

Bei der Evaluation wurden die Ergebnisse rahmenweise verglichen. Die Erkennung könnte voraussichtlich erheblich verbessert werden, wenn statt nur eines Rahmens auch die Ergebnisse der vorherigen Rahmen mit in die Erkennung einbezogen werden. Dies kann über eine Mittelwertbildung oder ein nachgeschaltetes Zustandsmodell erfolgen.

Die in Unterabschnitt 4.2.5 vorgeschlagene getrennte Vermessung von Geräten für das Training ist möglich, wie die funktionierende Erkennung zeigt. Aufgrund der PFC kommt es jedoch zu Abhängigkeiten zwischen den Netzspannungsbeeinflussungen, weshalb die getrennte Vermessung zu Fehlern in der Erkennung führt. Der Effekt einer PFC könnte in

Zukunft aber ebenfalls mathematisch bei der Modellierung und bei der Kombination von Geräten beachtet werden.

### 6.3.6 Fazit

Anhand der vorangegangenen Beschreibungen dieses Abschnitts kann festgestellt werden, dass Techniken der Signalverarbeitung und des automatischen Lernens für die Geräteeinbindung verwendet werden können, ohne dass große Folgefehler zu erwarten sind. Bei den Untersuchungen wurden Techniken der Signalverarbeitung in verschiedenen Aufgabenbereichen bestätigt. Darunter finden sich technisch einfache Lösungen, wie die Energieverbrauchszuordnung, aber auch komplexe Lösungen, wie die Mehrgeräteerkennung mit Verwendung von separaten Trainingsdaten einzelner Geräte.

Wie aus Unterabschnitt 6.3.2 deutlich wird, ist die Energieverbrauchsschätzung ein sinnvoller und in der Praxis anwendbarer Ansatz, der relativ einfach auf viele bereits vernetzte Geräte angewendet werden kann. Dadurch kann auf zusätzliche Energiemessgeräte vor jedem Gerät verzichtet werden. Die Analyse in Unterabschnitt 6.3.2 zeigt weiter, dass trotz der hohen variablen Lastkomponente durch die CPU eine sehr geringe normierte Gesamtabweichung erreicht werden konnte. Sofern Geräte keine variablen Lastkomponenten enthalten, sind mit der direkten Zuordnung des Energieverbrauchs aus Unterabschnitt 5.2.3 ähnlich präzise Schätzungen zu erwarten.

Am Beispiel der Zustandsidentifikation mittels der Schwellenwert-Klassifikation wird deutlich, dass auch mit technisch einfachen Erkennungstechniken eine Vielzahl von Geräten eingebunden werden kann (siehe Unterabschnitt 6.3.3). Durch das Modell aus Unterabschnitt 5.2.7 ist insbesondere eine Zuordnung Nicht-Programm-gesteuerter Geräte möglich. Wie in Unterabschnitt 6.3.4 anhand einer Waschmaschine gezeigt, kann aber auch der momentane Betriebsstatus über die Zeit von Programm-gesteuerten Geräten mit Hilfe einer entsprechenden Parametrisierung zugeordnet werden.

Bekannte Algorithmen, wie der hier verwendete DTW-Musterabgleich, erlauben zudem eine Erkennung des von der Nutzerin bzw. vom Nutzer gewählten Programmes, wie in Unterabschnitt 6.3.4 bestätigt wurde. Der DTW-Algorithmus kommt dabei auch mit sehr wenig Trainingsmaterial zurecht. Dies zeigt sich im Vergleich zu moderneren Erkennungsverfahren, wie z. B. den neuronalen Netzen, die für ihr Training eine große Anzahl von Trainingsdaten benötigen.

Mit der Mehrgeräteerkennung mittels künstlicher neuronaler Netze wurde eine Erkennung mehrerer Geräte implementiert und überprüft, ob Netzspannungsbeeinflussungen für die

Zustandsidentifikation genutzt werden können. Dabei gelang es, eine Struktur für das Training zu implementieren, die es ermöglicht ein Modell für die Erkennung aller Gerätekombinationen aus separaten Trainingsdaten zu trainieren, obwohl für das Training eigentlich alle Kombinationen aufgenommen werden müssten. Trotz dieser Vereinfachung konnten gute Erkennungsraten erreicht werden, was zeigt, dass die Struktur für das Training geeignet ist. Bei einigen Gerätekombinationen kam es zu Problemen, die auf eine gegenseitige Beeinflussung der Geräte untereinander zurückzuführen ist. Des Weiteren wurde durch die Evaluation deutlich, dass sich Netzspannungsbeeinflussungen grundsätzlich für die Zustandsidentifikation von elektronischen Geräten eignen. Weiterhin gibt es Geräteklassen, bei denen eine Identifizierung nicht möglich ist, da sie sehr wenige Netzspannungsbeeinflussungen auslösen. Insbesondere die Einbindung kostengünstiger elektronischer Geräte mit geringem Verbrauch, wie z. B. bei Ladegeräten, lohnt sich mit vielen Einbindungsverfahren aufgrund der überproportionalen zusätzlichen Kosten für individuell notwendige Sensoren oder Kommunikationsschnittstellen oft nicht. Weiterhin ist der Betriebszustand solcher Geräte aufgrund des geringen und oft variablen Verbrauchs schwer durch Einbindungsverfahren auf Grundlage der Identifikation anhand des Energieverbrauchs zu analysieren. Hier könnte zukünftig die Mehrgeräteerkennung ganzer Räume oder Haushalte anhand von Netzspannungsbeeinflussungen eine Lösung darstellen. Bei diesem Verfahren bleibt allerdings zu beachten, dass zwei identische Geräte nicht voneinander zu unterscheiden sind.

Zusammenfassend wird bestätigt, dass verschiedene Geräte unter Zuhilfenahme von unterschiedlichen Erkennungs- und Nachbildungsverfahren in das Energiemanagementsystem eingebunden werden können. Geräte können dabei in Kategorien untergliedert werden, damit sie jeweils mit geeigneten Techniken bzw. Modellen eingebunden werden können. Dieses Vorgehen wurde als Teil des Systemkonzepts bereits in der Einleitung zu Abschnitt 4.2 veranschaulicht. Als Erweiterung des Systemkonzepts könnten zusätzlich Programmgesteuerte sowie elektronische Geräte separat behandelt werden.

Bei der Konzeption und Implementierung der Erkennungs- und Nachbildungsverfahren wurden die Verfahren möglichst auf die wichtigsten Bestandteile beschränkt und eine zusätzliche Komplexität vermieden (vgl. Einleitung zu Kapitel 5). Damit wurden technisch einfache, aber solide Lösungen geschaffen, die nicht nur auf bestimmte Umgebungen oder Geräte fokussieren. Alle Lösungen könnten z. B. unter dem Einsatz von Techniken des automatischen Lernens weiter verbessert werden. Dazu wären jedoch große Datenbanken erforderlich, die im Rahmen dieser Arbeit nicht zur Verfügung standen. Mit der IoT-Softwareplattform wurde aber eine Lösung geschaffen, um systematisch solche Daten aufnehmen zu können.

## 6.4 Überprüfung der Zielfunktionalitäten

In Unterabschnitt 3.3.1 wurden Zielfunktionalitäten abgeleitet, die von der IoT-Softwareplattform unterstützt werden sollen. Diese wurden als exemplarische Anwendungen implementiert und in Abschnitt 5.3 vorgestellt. Aus diesen Beschreibungen wird deutlich, auf welche Weise die Funktionalitäten umgesetzt werden konnten. Die erfolgreiche Implementierung der Anwendungen erbringt somit den Nachweis, dass auf Basis der IoT-Softwareplattform alle Zielfunktionalitäten umgesetzt werden können.

Wie in Unterabschnitt 6.1.3 beschrieben, wurde die Funktionsfähigkeit der Anwendungen unter Verwendung der hierarchischen Kommunikationsstruktur innerhalb des Gesamtdemonstrators überprüft. Eine Visualisierung der Funktionsfähigkeit über den Umfang von Unterabschnitt 6.1.3 hinaus wird nicht als zielführend angesehen, da es sich bei vielen Anwendungen lediglich um einfache kausale Verknüpfungen handelt.

Aus den Implementierungen der Zielfunktionalitäten auf Basis der IoT-Softwareplattform werden einige Folgerungen deutlich:

- Alle Funktionalitäten können gemeinsam und in einem weiten Umfang in einem System realisiert werden. Die IoT-Softwareplattform enthält somit alle notwendigen Merkmale zur Ausführung der Funktionalitäten. Dadurch wird zudem die Vielseitigkeit der IoT-Softwareplattform und des Systemkonzepts deutlich.
- Ferner gelang es, die unterschiedlichsten Geräte über verschiedene Erkennungs- und Nachbildungsverfahren einzubinden, ohne dass spezifische Anpassungen oder Routinen in den Funktionalitäten erforderlich waren. Wie in Unterabschnitt 5.1.8 und 5.1.9 beschrieben, können die Funktionalitäten auf die gleiche Weise auf alle Geräte zugreifen.
- Aus den Beschreibungen der Anwendungen in Abschnitt 5.3 wird zudem deutlich, dass sich viele Funktionalitäten trotz ihres verteilten Charakters auf Basis der IoT-Softwareplattform mit sehr geringem Aufwand implementieren ließen. Dieses untermauert somit die Argumentation zur Erfüllung der Anforderung NF.6 in Unterabschnitt 6.2.4.

Die Funktionalitäten wurden bei der Konzeption auf ihre wichtigsten Funktionsmerkmale heruntergebrochen. Dadurch gelang eine leicht verständliche Darstellung der Funktionsprinzipien dieser Funktionalitäten und stellt damit ein weiteres hervorzuhebendes Ergebnis dieser Arbeit ist.

Die Untersuchung der tatsächlichen Anwendbarkeit der ausgearbeiteten Zielfunktionalitäten würde weit über die Zielsetzung der Konzeption eines Systemkonzepts sowie dessen Überprüfung hinausgehen. Analysen über die Kooperation der Funktionalitäten, über die

Auswirkungen von Folgefehlern aus den Geräteeinbindungen sowie über die Kombinationsfähigkeit zwischen den Funktionalitäten und den Geräteeinbindungen wurden somit im Rahmen dieser Arbeit nicht betrachtet. Aufgrund des erheblichen Umfangs werden diese Untersuchungen als eigenes zukünftiges Forschungsgebiet empfohlen.

## 6.5 Diskussion und Fazit zum Systemkonzept

In diesem Abschnitt werden die Auswahl und das Zusammenspiel der Komponenten des in Kapitel 4 beschriebenen Systemkonzepts anhand der zuvor dargestellten Untersuchungen diskutiert. Nachfolgend wird dazu zunächst auf die Defizite bisheriger Energiemanagementsysteme Bezug genommen und daraus Folgerungen für das Systemkonzept abgeleitet. Anschließend wird diskutiert, welche Kosten bei Verwendung des Systemkonzepts anfallen und ob eine großflächige Verbreitung möglich erscheint. Darauf aufbauend wird die Eignung der gewählten Elemente des Systemkonzepts diskutiert. Aus den Erkenntnissen dieser Arbeit werden danach Forderungen zur Verbesserung von zukünftigen Energiemanagementsystemen abgeleitet. Abschließend wird dargestellt, welche Implementierungen für einen Einsatz auch außerhalb der Forschung geeignet sind und damit ebenfalls ein Ergebnis dieser Arbeit darstellen.

### Überwindung der Defizite

Zur Demonstration wurde das System in verschiedenen realen Umgebungen eingesetzt. Wie in Unterabschnitt 6.1.4 festgestellt wurde, konnten dabei die beim Einsatz bisheriger Systeme auftretenden Defizite vermieden werden (siehe Abschnitt 3.2). So konnten mehrere Funktionalitäten in unterschiedlichen Umgebungen für verschiedene Stakeholder umgesetzt werden. Zusätzlich gelang es durch die Offenheit des Systems sowie der verwendeten Erkennungs- und Nachbildungsverfahren prinzipiell eine größere Anzahl unterschiedlicher Geräte einzubinden.

Durch die Möglichkeit, mit Hilfe der verteilten Anwendungen Daten lokal in den Umgebungen verarbeiten zu können und dem Einsatz der Regelsätze zur Filterung von Daten können Datenschutz- und Datensicherheitsanforderungen beachtet werden (siehe Unterabschnitt 6.2.4). Damit stellen Anforderungen an den Datenschutz und die Datensicherheit keinen Hinderungsgrund mehr für den Einsatz des Energiemanagementsystems dar.

Folglich kann festgestellt werden, dass das in Kapitel 1 genannte Ziel der Entwicklung eines offenen und flexiblen Systemkonzepts erreicht wurde.

### **Kostenreduzierung**

In Abschnitt 3.2 wurde die Problematik formuliert, dass die Neuinvestitionen bei einigen am Markt angebotenen Energiemanagementsystemen bei Annahme realistischer Amortisationszeiten vermutlich nicht rentabel sind. Nachfolgend soll dargelegt werden, welche Maßnahmen ergriffen wurden, um die Investitions- und Betriebskosten unter Verwendung des Systemkonzepts zu reduzieren.

Durch die Energiemanagement-Funktionalitäten können Energiekosten eingespart werden. Aufgrund der größeren Anzahl an Optimierungsansätzen dieser Funktionalitäten kann im Vergleich zu bisherigen Systemen ein zusätzliches Energieeinsparpotential gehoben werden. Weiterhin kann durch die Einbindung einer höheren Geräteanzahl ein zusätzliches Energieeinsparpotential gehoben werden, wenn auf diese ebenfalls Optimierungsansätze angewendet werden.

Wie aus den weiteren Funktionalitäten aus Abschnitt 4.6 deutlich wird, können zudem Investitions- und Betriebskosten reduziert werden, indem andere verwandte Infrastrukturen (siehe Unterabschnitt 2.3.1) gemeinsam in einem System umgesetzt werden. Mit zusätzlichen Funktionalitäten im Sinne des Smart Home, wie zum Beispiel durch die Steigerung des Komforts, kann für die Nutzerinnen und Nutzer ein zusätzlicher Mehrwert generiert werden. Dieser Mehrwert dürfte den Nutzerinnen und Nutzer zudem ein zusätzliches Investment bzw. zusätzliche laufende Kosten wert sein.

Die hierarchische Kommunikationsstruktur fordert im Vergleich zu zentralisierten Strukturen zusätzliche lokale Koordinatoren. Dies führt zu einer Erhöhung der Investitions- und Betriebskosten. Sofern vorhanden, können die Aufgaben der lokalen Koordinatoren zukünftig auch von anderen intelligenten Geräten mit übernommen werden. Hier bieten sich Internet- und WLAN-Router, Drucker, VoIP-Telefone, TV- und Radio-Geräte, aber auch festinstallierte Geräte, wie z. B. Kühlschränke, Backöfen oder Bewegungsmelder an.

Mit der Möglichkeit, Geräte mit den verschiedenen Techniken (siehe Abschnitt 5.2) einbinden zu können, wird eine weitere Kosteneffizienz fokussiert. Damit wird beispielsweise vermieden, dass an jedes Gerät ein Energiemess-Zwischenstecker angeschlossen werden muss. Die Beschaffung dieser Zwischenstecker ist relativ kostenintensiv. So sind Lösungen deutlich kosteneffizienter, die mit möglichst wenigen Messgeräten auskommen. Das kann durch die gleichzeitige Erkennung mehrerer Geräte oder auch durch die Schätzung von Messgrößen realisiert werden.

Wie aus diesen Argumenten deutlich wird, zielt das System und das zugrundeliegende Systemkonzept neben den technischen Vorteilen auch auf eine Erhöhung der Kosteneffizienz ab. Unter Betrachtung der Argumentation aus Abschnitt 3.2 lässt sich feststellen, dass mit diesem



System im Vergleich zu bisherigen Systemen ein deutlich größeres Energieeinsparpotential gehoben werden kann.

*Entsprechend kann abgeleitet werden, dass das primäre Ziel der Erhöhung der Energieeffizienz (siehe Kapitel 1) unter Verwendung des Systemkonzepts besser erreicht werden könnte, als mit bisherigen Systemen.*

### **Verbreitung & neuer Markt**

In der Einleitung zu Kapitel 3 wurde das Ziel formuliert, eine möglichst großflächige Verbreitung von Energiemanagementsystemen zu erreichen. Aus der Skalierungsfähigkeit (siehe Unterabschnitt 6.2.4) sowie den zuvor genannten Argumenten, kann geschlossen werden, dass auf Basis des Systemkonzepts für eine großflächige Verbreitung keine technischen Begrenzungen mehr existieren. Für eine erfolgreiche Verbreitung fehlt jedoch noch eine breite Unterstützung für die Einbindung von vernetzten und nicht-vernetzten Geräten. Auch werden weitere und ausgereifere Funktionalitäten benötigt. Durch die Erweiterungsfähigkeit der IoT-Softwareplattform können diese Funktionen durch Gerätehersteller, aber auch durch Softwarehersteller implementiert werden. Dabei könnten Softwarehersteller durch die Vermarktung von Software-Erweiterungen neue Märkte erschließen, ähnlich wie bei Apps für Smartphones und Tablets. Sinngemäß gilt dies auch für Gerätehersteller, die durch den Verkauf von eigenen Software-Erweiterungen zusätzliche Einnahmen generieren könnten. Weiterhin könnte den Kunden durch Produkte mit zusätzlichen Schnittstellen ein weiteres Kaufkriterium angeboten werden.

Da es bereits eine große Anzahl etablierter Produkte zum Energiemanagement auf verschiedenen Marktsegmenten gibt, dürfte eine Marktdurchsetzung der vorgestellten Implementierungen nur bedingt realisierbar sein. Die angestrebte Veröffentlichung der IoT-Softwareplattform als Open Source-Lösung soll dennoch ein Anreiz für Unternehmen und Privatpersonen sein, die Ideen des Systemkonzepts zu erproben und weiterzuentwickeln. Die einzelnen Elemente des Systemkonzepts stellen auch für sich genommen wertvolle Lösungen dar, die dazu verwendet werden könnten, bereits vorhandene Systeme weiter zu optimieren.

### **Auswahl der Grundelemente des Systemkonzepts**

Das Systemkonzept beruht bewusst nur auf wenigen grundlegenden Ideen, die konsequent verfolgt und umgesetzt wurden. Dabei wird im Vergleich mit den Systembeschreibungen des HEMS und zum industriellen Einsatz (siehe Abschnitt 2.2) deutlich, dass diese Grundelemente des Systemkonzepts bisher nicht grundsätzlich zum Energiemanagement gezählt werden. Die Auswahl dieser Grundelemente wird nachfolgend erörtert:

- Durch die Verwendung der **hierarchischen Kommunikationsstruktur** konnten im Vergleich mit zentralisierten Strukturen bestehende Hinderungsgründe in der Struktur bisheriger Systeme überwunden werden. Wie in Abschnitt 6.2 erläutert, ermöglicht Verwendung der Kommunikationsstruktur die Einhaltung von Datenschutz- und Datensicherheitsanforderungen. Damit behalten die Nutzerinnen und Nutzer die Datenhoheit über ihre Daten. Dies ist insbesondere für das Energiemanagement wichtig, da hier viele Daten anfallen, aus denen Rückschlüsse auf Betriebsgeheimnisse sowie Gewohnheiten der Menschen gezogen werden könnten (siehe Abschnitt 3.1). Die hierarchische Kommunikationsstruktur stellt damit ein geeignetes Konzept dar, das zudem auch viele weitere Vorteile in Hinsicht auf Skalierbarkeit und Ausfallsicherheit bietet (siehe Unterabschnitt 6.2.4). Wie in Unterabschnitt 6.2.3 diskutiert, fallen die zusätzlichen Latenzen bedingt durch die komplexere Struktur nicht ins Gewicht. Durch die Verkürzung der Kommunikationswege kann in der Regel sogar von deutlich geringeren Latenzen ausgegangen werden.
- Das Konzept der **verteilten IoT-Anwendungen** stellt eine flexible Lösung für die Verarbeitung von Daten in der hierarchischen Kommunikationsstruktur dar. Es konnte gezeigt werden, dass Anwendungen nach der verteilten Softwarekonzeption entworfen und eingesetzt werden können. Dabei konnten strukturbedingte Nachteile in der Software-Bereitstellung kompensiert werden (siehe Unterabschnitt 6.2.4). Aufgrund der lokalen Verarbeitung konnte zudem die Vermittlung kritischer Daten reduziert werden, sodass auch die Regelsätze zur Begrenzung der Weiterleitung von Datenressourcen im Vergleich zu ähnlichen Konzeptionen rudimentärer gestaltet werden konnten (vgl. die in Unterabschnitt 2.3.5 vorgestellten Gemeinschaftshierarchien).
- Das Konzept der **transparenten Kommunikation** ermöglicht den Anwendungen zudem einen unkomplizierten Datenaustausch, trotz der zu bisherigen Systemen wesentlich aufwendigeren Kommunikationsstruktur. Dabei stellt sich die Verwendung der zwei Kommunikationsarten, also des Publish-Subscribe-Entwurfsmusters und der Möglichkeit zum Aufruf von entfernten Funktionen (RPC) als sehr sinnvoll heraus. So können Anwendungsfälle zur nicht-adressierten Veröffentlichung von z. B. Messreihen und Ereignissen getrennt von Anwendungsfällen zum gezielten Aufruf z. B. von Schaltfunktionen behandelt werden (siehe auch Unterabschnitt 6.2.3). Die in Unterabschnitt 2.3.4 beschriebenen Protokolle greifen dabei nur auf eine Kommunikationsart zurück und können damit einen der beiden Anwendungsfälle nicht ohne Umwege abbilden.
- Durch die Definition des **notwendigen Funktionsumfangs** (siehe Unterabschnitt 4.1.1) gelang es, Geräte unterschiedlicher Hersteller über verschiedene Infrastrukturen bzw. über verschiedene Techniken gleichartig einzubinden, sodass diese in höheren

Schichten auf die gleiche Weise angesprochen werden konnten. Wie bereits dargestellt, konnte dadurch eine breite Auswahl an Lösungen implementiert und erprobt werden. Durch diese Definition wurde zudem das Verständnis gewonnen, welche Daten und Funktionen von Geräten benötigt werden und wie diese ggfs. nachgebildet werden können. Die Gerätemodellierung kann dazu beitragen, fehlende Messdaten nachzubilden. Weiterhin ist die Möglichkeit gegeben durch die Zustandsidentifikation den Betriebsstatus der Geräte nachzubilden. Solche Verfahren sind zwar bekannt, wurden bisher aber lediglich im Zusammenhang mit Energieinformationssystemen beobachtet (siehe Unterabschnitt 2.2.3). Insbesondere die Energieverbrauchsschätzung ist hier hervorzuheben, da sie kostensparend realisiert werden kann und dennoch hohe Genauigkeiten erreicht (siehe Unterabschnitt 6.3.2). Wie in Unterabschnitt 6.3.6 gefolgert wurde, sind die verschiedenen Verfahren für die Einbindung der jeweiligen Gerätearten unterschiedlich geeignet. Deshalb hat sich die Forderung des Systemkonzepts, verschiedene Anbindungsmöglichkeiten für eine möglichst hohe Anzahl unterschiedlicher Geräte vorzusehen, als sinnvoll und kostendämpfend erwiesen.

- Anhand der Implementierung der **Zielfunktionalitäten** wird deutlich, dass alle gewünschten Energiemanagement-Maßnahmen in einem System realisiert werden können (siehe Abschnitt 6.4). Trotz des verteilten Charakters der hierarchischen Kommunikationsstruktur konnten alle definierten Funktionalitäten in einem weiten Umfang bei gleichzeitig verhältnismäßig geringem Implementierungsaufwand realisiert werden. So wurden sowohl das automatisierte Energiemanagement und auch Nutzer-Unterstützungen adressiert. Durch die Realisierung der Funktionalitäten jenseits des Energiemanagements konnte zudem bestätigt werden, dass das Energiemanagement gemeinsam mit anderen Themenfeldern implementiert werden kann und die anvisierte Ausnutzung von Synergieeffekten möglich ist. Zudem konnte gezeigt werden, dass die IoT-Softwareplattform somit alle notwendigen Merkmale zur Ausführung der Funktionalitäten enthält.

Deutlich wird, dass alle Grundelemente des Systemkonzepts ihre Eignung zeigen konnten. Dabei wird erst durch die Kombination dieser Grundelemente auf Basis des gewählten Schichtenmodells die Flexibilität und Funktionsfähigkeit des Systems ermöglicht. Insgesamt konnte auf Basis der Grundelemente ein System erschaffen werden, das die in Abschnitt 3.1 dargelegte Vision in allen Punkten erfüllt.

Unter bestimmten Voraussetzungen können die Grundelemente auch einzeln auf bestehende Energiemanagementsysteme adaptiert werden. Dies gilt insbesondere für die Verfahren zur Geräteeinbindung. Weiterhin können bei vorhandener Zugriffsfähigkeit auf die Geräte auch einzelne Funktionalitäten in vorhandenen Systemen implementiert werden. Zusätzlich

kann in bestehenden Systemen auch die hierarchische Kommunikationsstruktur Anwendung finden, indem eine Vernetzungsmöglichkeit zwischen zwei Instanzen geschaffen wird. Ohne die Verwendung des Konzepts der verteilten Anwendungen und der Erweiterungsmöglichkeiten eines offenen Systems bleibt es jedoch schwierig, die Funktionalitäten umzusetzen und gleichzeitig konkreten Anforderungen an den Datenschutz und die Datensicherheit gerecht zu werden.

### **Forderungen für zukünftige Energiemanagementsysteme**

Nachfolgend wird der Frage nachgegangen, welche Konsequenzen und Forderungen für die Entwicklung zukünftiger Energiemanagementsysteme aus dieser Arbeit abgeleitet werden können:

- Nutzer-unterstützende und automatisierte Energiemanagementsysteme (siehe Abschnitt 2.2) sollten nicht weiter getrennt voneinander betrachtet werden, da sich deutliche Synergieeffekte ergeben.
- Energiemanagementsysteme sollten als integrale Bestandteile anderer Infrastrukturen insbesondere aus den Bereichen des Smart Home und der Industrie 4.0 realisiert werden, da sich auch hier Synergieeffekte ergeben. In dieser Arbeit wurde durch die Implementierung von Funktionalitäten jenseits des Energiemanagements gezeigt, dass die Kombination der Infrastrukturen möglich ist. Durch die Vermeidung paralleler Infrastrukturen konnte ein Beitrag für eine Reduzierung der Investitions- und Betriebskosten geleistet werden. Erheblich sinnvoller wäre eine einheitliche Infrastruktur, auf der Anwendungen beliebiger Themenbereiche ausgeführt werden könnten.
- Bereits vernetzte Geräte sollten über ihre Netzwerkschnittstellen Informationen zu ihrem (typischen) Energieverbrauch mitteilen. Wie in dieser Arbeit gezeigt, können solche Geräte direkt und kosteneffizient eingebunden werden. Hilfreich wäre eine gesetzliche oder normative Verpflichtung der Gerätehersteller, ihre Geräte mit offenen Schnittstellen auszurüsten. Dies könnte eine zukünftige Voraussetzung zur Erfüllung bestimmter Energieeffizienz-Label sein. Sofern keine zusätzlichen Sensoren für die Bereitstellung bestimmter Informationen verbaut werden sollen, könnten hier die Techniken aus Unterabschnitt 5.2.2 und folgende eingesetzt werden. Eine solche Einbindung wurde für Computer anhand der Energieverbrauchsschätzung aus Unterabschnitt 5.2.4 gezeigt.
- In Energiemanagementsysteme sollten mehr Verfahren aus der Signalverarbeitung und -erkennung Einzug halten. Wie in Abschnitt 6.3 gezeigt wurde, konnten erfolgreiche Einbindungen von Geräten für die Ermittlung des Betriebsstatus und des Energieverbrauchs realisiert werden.

- Auch bestehende nicht-vernetzte Geräte sollten eingebunden werden. Dazu sollten u. a. die oben benannten Verfahren verwendet werden.
- Anforderungen an den Datenschutz und die Datensicherheit sollten durch die Energiemanagementsysteme beachtet werden können. In dieser Arbeit wurde dieses auf Basis der Konzepte der hierarchischen Kommunikationsstruktur sowie der verteilten Anwendungen gezeigt.
- Verschiedene Energiemanagement-Funktionalitäten sollten parallel ausgeführt werden können. Auch eine Ausführung über mehrere Umgebungen hinweg sollte dabei möglich sein. Beides wurde in dieser Arbeit demonstriert.

### Verwendbare Implementierungen

Die Eignung der verschiedenen Verfahren und Konzepte wurde anhand der durchgeführten Implementierungen bestätigt. Bis auf wenige Ausnahmen wurden alle Implementierungen nicht nur für die Evaluation, sondern auch für den Live-Betrieb konzipiert und umgesetzt. Nachfolgend wird dargelegt, welche Implementierungen sich für einen Einsatz auch außerhalb der Forschung eignen. Die entstandenen Implementierungen sind somit ein weiteres bedeutendes Ergebnis dieser Arbeit.

Mit der **IoT-Softwareplattform** wurde eine IoT-fähige Basis entwickelt, die viele neue vernetzte Anwendungen ermöglicht und sich mit der Leistungsfähigkeit weitverbreiteten quelloffenen Softwareprodukten zur Datenvermittlung messen kann. Auf die Erfüllung der Anforderungen wurde besonderen Wert gelegt, wodurch auch eine produktive Einsatzfähigkeit gegeben ist (siehe Unterabschnitt 6.2.5).

Viele der **Anwendungen**, die für die IoT-Softwareplattform entwickelt wurden, können in der Praxis eingesetzt werden. Die Auflistung in Unterabschnitt 5.1.7, sowie die in Abschnitt 5.2 und 5.3 erwähnten Anwendungen verdeutlichen die große Zahl der Implementierungen. Dazu zählen:

- die zahlreichen Anbindungen von Infrastrukturen und Bussystemen (siehe Unterabschnitt 5.1.7),
- die Möglichkeiten der Geräteeinbindung, wie z. B. das Modell für die Schwellenwert-Klassifikation (siehe Unterabschnitt 5.2.7),
- die Energieverbrauchsermittlung, z. B. von Desktop-Computern (siehe Unterabschnitt 5.2.4),
- die grafischen Nutzeroberflächen, wie z. B. das Energiecockpit (siehe Unterabschnitt 5.3.1) und

- die automatisierten Energiemanagementanwendungen, wie z. B. die Abstimmung von Geräten (siehe Unterabschnitt 5.3.2) oder die Anwendung zur Eigenverbrauchsoptimierung (siehe Unterabschnitt 5.3.4).

Damit steht ein umfangreiches Softwarepaket für das Energiemanagement zur Verfügung, das hohe Freiheitsgrade für die Realisierung gewünschter Funktionen ermöglicht. Mit den Anwendungen können viele Funktionalitäten für das Energiemanagement, für das Smart Home und für die Industrie 4.0 realisiert werden.

## 7 Zusammenfassung und Ausblick

Das Ziel dieser Arbeit war die Erstellung eines universellen und flexiblen Systemkonzepts für Energiemanagementsysteme auf Basis von verteilten Systemen nach einem allgemeinen Nutzer-zentrierten Ansatz unter Berücksichtigung der Ideen des Internet der Dinge.

Im Anschluss an eine Analyse des Standes der Technik in Kapitel 2 und einer daraus in Kapitel 3 abgeleiteten Vision für ein möglichst weit verbreitetes, optimales Energiemanagement wurden die Defizite existierender Lösungen festgestellt. Aufbauend auf der Vision wurden Ziele und Anforderungen für zukünftige Energiemanagementsysteme definiert.

Kapitel 4 diente der Entwicklung eines die Vision umsetzenden Systemkonzepts. Das Systemkonzept besteht aus verschiedenen Komponenten und Verfahren und wurde in drei Schichten aufgeteilt:

- Die unterste Schicht dient der Einbindung von vernetzten und nicht-vernetzten Geräten. Hierzu wurde ein notwendiger Funktionsumfang für die Einbindung von Geräten definiert. Um fehlende Daten und Funktionen nachzubilden, wurden Verfahren auf Basis der Signalverarbeitung und -erkennung vorgeschlagen.
- Die mittlere Schicht ist für die Kommunikation verantwortlich. Dazu wurde eine hierarchische Kommunikationsstruktur zur lokalen Vermittlung und Verarbeitung von Daten mit Hilfe von verteilten Anwendungen beschrieben. Die Kommunikationsstruktur erlaubt Anforderungen an den Datenschutz und die Datensicherheit einzuhalten und dennoch eine Kooperation aller Elemente zu ermöglichen.
- Die obere Schicht ermöglicht die Ausführung von Funktionalitäten des Energiemanagements. Hierzu wurden verschiedene Energiemanagement-Funktionalitäten identifiziert und deren Funktionsweise in der hierarchischen Struktur dargelegt. Zusätzlich wurden auch Funktionalitäten jenseits des Energiemanagements betrachtet.

Zur Erprobung des Systemkonzepts wurden die folgenden Implementierungen erstellt und in Kapitel 5 beschrieben:

- Mit der sogenannten IoT-Softwareplattform wurde eine universelle Basis in der Skriptsprache JavaScript geschaffen, mit der verteilte Anwendungen ausgeführt werden

können. Lokale Daten werden in sogenannten Datenressourcen gehalten und können über ausgearbeitete Basismethoden angesprochen werden. Die Basismethoden beruhen sowohl auf dem Publish-Subscribe-Entwurfsmuster als auch auf Konzepten zum entfernten Aufruf von Methoden (engl. Remote Procedure Calls, RPC). Für die Kommunikation in der hierarchischen Kommunikationsstruktur wurde ein auf diesen Basismethoden aufsetzendes Protokoll definiert. Auf Basis der IoT-Softwareplattform wurden viele Anwendungen ebenfalls in JavaScript umgesetzt, die die Softwareplattform um verschiedenste Funktionen erweitern, so z. B. den Zugriff auf andere Kommunikationsprotokolle, Feldbusse und Sensoren. Nutzerschnittstellen können auf Grundlage des Model-View-ViewModel-Entwurfsmusters erstellt werden.

- Für die Einbindung von vernetzten und nicht-vernetzten Geräten wurden verschiedene Geräteeinbindungen für die IoT-Softwareplattform geschaffen. Hier wurden unter anderem Techniken der Signalverarbeitung und -erkennung angewandt. Diese Einbindungsverfahren wurden in JavaScript oder auf Basis der numerischen Berechnungsumgebung MATLAB entwickelt. Die Energieverbrauchsschätzung ermöglicht es, den Energieverbrauch von bereits vernetzten Geräten anhand interner Statusdaten zu schätzen. Für Desktop-Computer steht eine Implementierung zur Verfügung, die den Energieverbrauch anhand der CPU- und GPU-Last abschätzt. Auch verschiedene Verfahren zur Zustandsidentifikation von nicht-vernetzten Geräten wurden entwickelt und evaluiert. Darunter finden sich einfache Zuordnungsverfahren anhand von Schwellenwerten, Verfahren zur Mustererkennung, wie dem DTW-Algorithmus, aber auch Verfahren zur Mehrgeräteerkennung mittels künstlicher neuronaler Netze.
- Für alle konzipierten Funktionalitäten wurden jeweils eine oder mehrere Anwendungen für die IoT-Softwareplattform implementiert. Darunter befinden sich Nutzer-unterstützende Funktionalitäten, wie z. B. die Smartphone-Anwendung „Meine Energie“, die direkte Rückmeldungen der Gerätenutzungen bereitstellt. Weiterhin ist auf die Display-Anwendung „Energiecockpit“ zu verweisen, die Expertinnen und Experten sowie erfahrenen Anwenderinnen und Anwendern einen Überblick über den Zustand ihrer Geräte zur Verfügung stellt. Zusätzlich finden sich Funktionalitäten für automatisierte Energiemanagement-Anwendungen, die eine Abstimmung von Geräten untereinander realisieren oder auf variable Tarife sowie auf Lastspitzen reagieren können.

In Kapitel 6 wurden der Einsatz und die Evaluation des Gesamtsystems dargelegt. Die Einsatzfähigkeit wurde in verschiedenen realen Umgebungen erprobt und demonstriert. So wurden an der TU Braunschweig verschiedene Umgebungen erschlossen und im Sinne der hierarchischen Kommunikationsstruktur zu dem Gesamtdemonstrator „Real-Life-Lab



Campus 2020“ zusammengeführt. Im Verlauf der Erprobung wurden die Komponenten des Systemkonzepts einzeln untersucht:

- Für die Evaluation der IoT-Softwareplattform wurde gezeigt, auf welche Weise die zuvor ausgearbeiteten Anforderungen erfüllt wurden. Durch die hierarchische Kommunikationsstruktur ergeben sich demnach im Vergleich zu zentralisierten Strukturen Vorteile, wie eine bessere Skalierungsfähigkeit und eine höhere Ausfallsicherheit. Auch wurde anhand der Latenzen und der Datenraten das Systemverhalten der IoT-Softwareplattform aufgezeigt. Hierbei wurde die hohe Leistungsfähigkeit der Plattform im Vergleich zu weitverbreiteten quelloffenen Softwareprodukten zur Datenvermittlung unter Beweis gestellt.
- Durch den definierten notwendigen Funktionsumfang können Geräte unterschiedlicher Hersteller über verschiedene Infrastrukturen bzw. über verschiedene Techniken gleichartig eingebunden werden. Die dazu implementierten Techniken der Signalverarbeitung und -erkennung wurden ebenfalls analysiert. Die Energieverbrauchsschätzung für Computer erreicht eine sehr hohe Genauigkeit. Daraus wird gefolgert, dass die Energieverbrauchsschätzung eine geeignete Lösung ist und auch auf andere vernetzte Geräte ohne dynamische Lastkomponenten angewendet werden kann. Die Zustandsidentifikation anhand eines Schwellenwert-Klassifikators unter Verwendung eines Modells eignet sich besonders zur Einbindung von Geräten, die keiner Regelung oder Programmsteuerung unterliegen. Mit einzelnen Gerätemodellen können aber auch andere Geräte eingebunden werden. Mittels eines Algorithmus zum Musterabgleich konnte zudem eine einwandfreie Unterscheidung von verschiedenen Programmabläufen eines Gerätes realisiert werden. Der dazu verwendete DTW-Algorithmus ist dafür geeignet und benötigt nur wenige Trainingsdaten. Darüber hinaus konnte eine Mehrgeräteerkennung anhand von Netzspannungsbeeinflussungen erfolgreich eingesetzt werden. Sie eignet sich demnach insbesondere für elektronische Geräte. Insgesamt konnte gezeigt werden, dass der Einsatz mehrerer verschiedener Verfahren zur Geräteeinbindung sinnvoll und rentabel ist. Dieses an das Systemkonzept gestellte Ziel wird damit erfüllt.
- Anhand der Erprobung des Gesamtsystems konnte festgestellt werden, dass sich das angestrebte Energiemanagement realisieren lässt. Es konnte zudem aufgezeigt werden, dass die gewünschten Funktionalitäten auch in der verteilten Struktur gegeben sind und mit geringem Aufwand implementiert werden konnten.

Abschließend lässt sich feststellen, dass das genannte Ziel der Entwicklung eines universellen und flexiblen Systemkonzepts erreicht wurde und der anvisierte großflächige Einsatz von Energiemanagementsystemen technisch möglich ist. Die identifizierten Defizite vorhandener

Systeme konnten auf Grundlage des Systemkonzepts durch die Kombination verschiedener überprüfter Komponenten und Verfahren überwunden werden. Anforderungen an den Datenschutz und an die Datensicherheit konnten dabei durch lokale Verarbeitungsinstanzen mit ihren betreffenden Umgebungen verknüpft und eingehalten werden. Die dazu gewählte hierarchische Kommunikationsstruktur stellt sich als geeignetes Konzept dar, das zudem weitere Vorteile hinsichtlich der Skalierungsfähigkeit und Ausfallsicherheit bietet. Weiterhin konnte gezeigt werden, dass aufgrund der Kombination von verschiedenen Verfahren potentiell ein im Vergleich zu vorhandenen Systemen höheres Energieeinsparpotential gehoben werden kann. Zudem können durch die Ausnutzung von Synergieeffekten mit ähnlichen Infrastrukturen Investitions- und Betriebskosten gesenkt werden. Die entstandene IoT-Softwareplattform ist ein weiteres Ergebnis dieser Arbeit und eignet sich aufgrund ihrer Leistungsfähigkeit auch für einen produktiven Einsatz außerhalb der Forschung.

Viele der in dieser Arbeit entwickelten Konzepte und Verfahren können weiterentwickelt werden. Damit ergeben sich weitere mögliche neue Forschungsgebiete:

- In dieser Arbeit wurde ausschließlich auf die Energieform der elektrischen Energie eingegangen. Viele der Erkenntnisse könnten jedoch auch auf andere Energieformen und Verbrauchsmedien übertragen werden.
- Eine über den Umfang dieser Arbeit hinausgehende Erprobung des Energiemanagements auf Basis des Systemkonzepts, insbesondere in weiteren Umgebungen und Branchen, wäre sinnvoll.
- Auch eine nähere Untersuchung des Zusammenspiels mit den Themengebieten des Smart Home und der Industrie 4.0 wäre verfolgenswert.
- Die Verbesserung und Analyse der Eigenverbrauchsoptimierung bei Prosumern in verteilten Systemen stellt ein neues Forschungsgebiet dar, dessen Untersuchung sinnvoll wäre.

Zusätzlich könnten einige der vorgestellten Lösungen weiter betrachtet werden:

- Die Energieverbrauchsschätzung und ihre Modellierung könnte auf weitere bereits vernetzte Gerätearten angewendet werden, wie z. B. Fernseher, Drucker, Netzwerkspeicher, Router und Telefonanlagen. Besonders in Büroumgebungen mit vielen Geringverbrauchern kann diese Technik von Interesse sein.
- Das Konzept der hierarchischen Kommunikationsstruktur unter Verwendung von verteilten Anwendungen könnte auf andere Bereiche übertragen werden, in denen sensible Informationen verarbeitet werden. So wäre z. B. eine Anwendung in Krankenhäusern

denkbar, in denen sensible Informationen, wie z. B. Patientendaten, Laborergebnisse und direkte Messdaten von Patienten anfallen. Entsprechend würden sich Anwendungen am Krankenbett, im Krankenzimmer bzw. im Operationssaal, auf der Krankenstation und im gesamten Krankenhaus ergeben.

Die Erkenntnisse dieser Arbeit bieten eine Grundlage dafür, die bearbeiteten Themenfelder aufzugreifen und im wissenschaftlichen Kontext weiter zu bearbeiten. Insbesondere für die Einhaltung des Datenschutzes im Bereich des Internet der Dinge werden relevante Ansätze vorgestellt.



## Begriffs- und Abkürzungsverzeichnis

Die in diesem Verzeichnis aufgeführten Begriffe werden im Text bei der ersten Verwendung kursiv hervorgehoben.

|  |         |
|--|---------|
| <i>Advanced Message Queue Protocol</i> (engl., AMQP) . . . . .   | 39      |
| <i>Altersgerechtes bzw. umgebungsunterstütztes Wohnen</i> (engl. Ambient Assisted Living, AAL) . . . . . | 29      |
| <i>Ameisenalgorithmus</i> (engl. Ant Colony Optimization, ACO) . . . . .                                 | 25      |
| <i>Anzeigeinterface</i> (engl. User Interface, UI) . . . . .   | 88      |
| <i>Approximative dynamische Optimierung</i> (engl. Approximate dynamic programming, ADP) . . . . .       | 25      |
| <i>Auto Discovery</i> (engl. für automatische Erkennung) . . . . .                                       | 45      |
| <i>Automatisierte Energieverbrauchsoptimierung</i> . . . . .   | 23      |
| <i>Automatisiertes Energiemanagementsystem</i> . . . . .   | 20      |
| <i>Big-Data-Analysen</i> . . . . .   | 34      |
| <i>Binäre Partikelschwarmoptimierung</i> (engl. Binary Particle Swarm Optimization, BPSO) . . . . .      | 25      |
| <i>Client-Server-Kommunikation</i> . . . . .   | 33      |
| <i>Clouddienste</i> . . . . .  | 34      |
| <i>Cloud</i> (engl. für Wolke) . . . . .   | 34      |
| <i>Constrained Application Protocol</i> (engl., CoAP) . . . . .  | 39      |
| <i>Cyber-physische Systeme</i> (engl. Cyber Physical Systems, CPS) . . . . .                             | 31      |
| <i>Datenressourcen</i> . . . . .   | 99, 100 |
| <i>Datenschutz</i> . . . . .   | 30      |
| <i>Datensicherheit</i> . . . . .   | 30      |
| <i>Demand Response</i> (engl. für Lastmanagement, DR) . . . . .  | 13      |
| <i>Demand Side Management</i> (engl. für Management auf der Verbraucherseite, DSM) . . . . .             | 8       |
| <i>Dienstgüte</i> (engl. Quality of Service, QoS) . . . . .  | 39      |
| <i>Dienst-orientierte Plattformen</i> . . . . .  | 37      |
| <i>Digitale Repräsentation</i> . . . . .   | 60      |
| <i>Direkte Kommunikation</i> . . . . .   | 33      |
| <i>Disaggregation</i> . . . . .  | 78      |
| <i>Dynamische Zeitanpassung</i> (engl. <i>Dynamic Time Warping</i> , DTW) . . . . .                      | 124     |

|   |           |
|---|-----------|
| <i>Echtzeit</i> . . . . .   | 58        |
| <i>Echtzeitanforderungen</i> . . . . .  | 58        |
| <i>Echtzeitrückmeldungen</i> . . . . .  | 146       |
| <i>Echtzeit-Tarifgestaltung</i> (engl. Real-Time Pricing, RTP) . . . . .                            | 13        |
| <i>Economic Demand Response</i> . . . . .   | 13        |
| <i>Eigenverbrauchsoptimierung</i> . . . . .   | 19        |
| <i>Emergency Demand Response</i> . . . . .  | 14        |
| <i>Endverbraucher</i> . . . . .   | 9         |
| <i>EnEff Campus 2020</i> . . . . .  | 142       |
| <i>Energieeffizienz</i> . . . . .   | 1, 11     |
| <i>Energiemanagement</i> . . . . .  | 1         |
| <i>Energiemanagementsystem</i> . . . . .  | 2, 19, 28 |
| <i>Energiesparen</i> . . . . .  | 2         |
| <i>Energieverbrauchsermittlung</i> . . . . .  | 67        |
| <i>Energievertriebsunternehmen</i> . . . . .  | 9         |
| <i>Energiewende</i> . . . . .   | 1         |
| <i>Energy Information System</i> (engl., EIS) . . . . .   | 26        |
| <i>Energy Management System</i> (engl., EMS) . . . . .  | 21        |
| <i>Ereignis-basierte Plattformen</i> . . . . .  | 37        |
| <i>Events</i> (engl. für Ereignis) . . . . .  | 37        |
| <i>Extensible Markup Language</i> (engl., XML) . . . . .  | 38        |
| <i>Extensible Messaging and Presence Protocol</i> (engl., XMPP) . . . . .                           | 39        |
| <i>Fog-Computing</i> (engl. Fog für Nebel) . . . . .  | 35        |
| <i>Funk-Identifikations-Chips</i> (engl. Radio-Frequency Identification, RFID) . . . . .            | 33        |
| <i>Funk-Sensor-Netze</i> (engl. Wireless Sensor Networks, WSN) . . . . .                            | 33        |
| <i>Funkzugangsnetz</i> (engl. Radio Access Network, RAN) . . . . .                                  | 35        |
| <i>Fuzzy-Logik-Steuerung</i> (engl. Fuzzy Logic Control, FLC) . . . . .                             | 25        |
| <i>Gebäudeautomation</i> . . . . .  | 29        |
| <i>Gebäudeleittechnik</i> (GLT) . . . . .   | 29        |
| <i>Gemischte ganzzahlige lineare Optimierung</i> (engl. Mixed Integer Linear Programming) . . . . . | 25        |
| <i>Genetischer Algorithmus</i> (engl. Genetic Algorithm, GA) . . . . .                              | 25        |
| <i>Geräte</i> . . . . .   | 2, 59     |
| <i>Gerätedisaggregation</i> . . . . .   | 27        |
| <i>Herkömmliche Geräte</i> . . . . .  | 54        |
| <i>Hidden-Markoff-Modelle</i> (engl. Hidden Markov Model, HMM) . . . . .                            | 27, 77    |
| <i>Home Area Network</i> (engl., HAN) . . . . .   | 22        |
| <i>Home Energy Management System</i> (engl., HEMS) . . . . .  | 21        |

|  |           |
|--|-----------|
| <i>Hop</i> . . . . .   | 154       |
| <i>Hyper Text Transfer Protocol</i> (engl., HTTP) . . . . .                          | 38        |
| <i>Industrie 4.0</i> . . . . .   | 5, 31     |
| <i>Industrieautomation</i> . . . . .   | 29        |
| <i>Intelligente Gebäudesteuerung</i> (engl. Smart Building Control) . . . . .        | 29        |
| <i>Intelligente Geräte</i> . . . . .   | 4, 36, 53 |
| <i>Intelligente IoT-Geräte</i> (engl. Smart IoT Devices) . . . . .                   | 36        |
| <i>Intelligenter Stromzähler</i> (engl. Smart Meter) . . . . .                       | 22        |
| <i>Intelligente Vernetzung</i> . . . . .   | 5         |
| <i>Internet der Dinge</i> (engl. Internet of Things, IoT) . . . . .                  | 5, 30, 32 |
| <i>Internet Protokoll</i> (IP) . . . . .   | 155       |
| <i>Interne Zustände</i> . . . . .  | 65        |
| <i>Interoperabilität</i> . . . . .   | 30        |
| <i>IoT-Geräte</i> . . . . .  | 63        |
| <i>IoT-Gerätetrepräsentation</i> . . . . .   | 69        |
| <i>IoT-Softwareplattform</i> . . . . .   | 97        |
| <i>IPv6 over Low power Wireless Personal Area Network</i> (engl., 6LoWPAN) . . . . . | 39        |
| <i>JavaScript Object Notation</i> (engl, JSON) . . . . .                             | 38        |
| <i>Jitter</i> . . . . .  | 152       |
| <i>Künstliche neuronale Netze</i> (engl. Artificial Neural Network, ANN) . . . . .   | 25, 77    |
| <i>Kommunikationsmedien</i> . . . . .  | 32        |
| <i>Kompatibilitätsschicht</i> . . . . .  | 63        |
| <i>Lastspitze</i> . . . . .  | 18        |
| <i>Lastspitzenreduktion</i> . . . . .  | 18        |
| <i>Lastverschiebung</i> (engl. Load Shifting) . . . . .                              | 18        |
| <i>Latenz</i> . . . . .  | 151       |
| <i>Leipziger Strombörse EEX</i> (engl. European Energy Exchange) . . . . .           | 9         |
| <i>Lokaler Koordinator</i> (LK) . . . . .  | 83        |
| <i>Lokales Netzwerk</i> (engl. Local Area Network, LAN) . . . . .                    | 22        |
| <i>Maßnahmen des Energiemanagements</i> . . . . .                                    | 2, 17     |
| <i>Market Demand Response</i> . . . . .  | 13        |
| <i>Maximum a-posteriori</i> (MAP) . . . . .  | 77        |
| <i>Merkmalsextraktion</i> . . . . .  | 76        |
| <i>Message Queuing Telemetry Transport</i> (engl., MQTT) . . . . .                   | 38        |
| <i>Middleware</i> . . . . .  | 37        |
| <i>(Mobiles) Cloud-Computing</i> . . . . .   | 34        |
| <i>(Mobiles) Edge-Computing</i> . . . . .  | 35        |

|   |           |
|---|-----------|
| <i>Multimedia-Geräte</i> . . . . .  | 29        |
| <i>Netzbetreiber</i> . . . . .  | 9         |
| <i>Netznutzungsentgelte</i> . . . . .   | 9         |
| <i>Nicht-vernetzte Geräte</i> . . . . .   | 59        |
| <i>Nonintrusive Appliance Load Monitoring</i> (engl., NIALM) . . . . .  | 27        |
| <i>Nonintrusive Load Monitoring</i> (engl., NILM) . . . . .   | 27, 78    |
| <i>Nutzer-unterstützendes Energiemanagementsystem</i> . . . . .   | 19        |
| <i>Operating System for Internet of Things Applications</i> (engl. für System zur Ausführung<br>von Anwendungen des Internet der Dinge) . . . . .               | 98        |
| <i>PDCA-Kreislauf</i> . . . . .   | 27        |
| <i>Peer-to-Peer-Kommunikation</i> (P2P) . . . . .   | 33        |
| <i>Physical Demand Response</i> . . . . .   | 14        |
| <i>Polling</i> (engl. für Abfragen) . . . . .   | 159       |
| <i>Powerline-Kommunikation</i> (engl. Power Line Communication, PLC) . . . . .  | 32        |
| <i>Prädiktive Instandhaltung</i> . . . . .  | 29        |
| <i>Prosumer</i> . . . . .   | 10        |
| <i>Publish-Subscribe-Entwurfsmuster</i> . . . . .   | 37        |
| <i>Qualität der Erfahrung</i> (engl. Quality of Experience, QoE) . . . . .  | 24        |
| <i>Really Simple Syndication</i> (engl. RSS) . . . . .  | 38        |
| <i>Representational State Transfer</i> (engl., REST) . . . . .  | 38        |
| <i>Schnelle Fourier-Transformation</i> (engl. Fast Fourier Transformation, FFT) . . . . .   | 127       |
| <i>Schnittstelle zum Aufruf entfernter Funktionen</i> (engl. Remote Procedure Call, RPC) . . . . .  | 38        |
| <i>Schwachlasterhöhung</i> (engl. Valley Filling) . . . . .   | 18        |
| <i>Semi-statische Geräte</i> . . . . .  | 54        |
| <i>Sicherheitsüberwachung</i> . . . . .   | 29        |
| <i>Simple Object Access Protocol</i> (engl., SOAP) . . . . .  | 38        |
| <i>Smart City</i> (engl. für intelligente Stadt) . . . . .  | 32        |
| <i>Smart Grid</i> (engl. für intelligentes Netz) . . . . .  | 15        |
| <i>Smart Home Energy Management</i> (engl., SHEM) . . . . .   | 21        |
| <i>Smart Home</i> (engl. für intelligentes Zuhause) . . . . .   | 5, 29, 31 |
| <i>Smartplugs</i> . . . . .   | 22        |
| <i>Spitzenlastreduktion</i> (engl. Peak Clipping) . . . . .   | 18        |
| <i>Tarifgestaltung bei kritischen Lastspitzen</i> (engl. Critical Peak Pricing, CPP) . . . . .  | 13        |
| <i>Universal Plug and Play</i> (engl., UPnP) . . . . .  | 39        |
| <i>Verband Europäischer Übertragungsnetzbetreiber für Strom</i> (engl. European Network of<br>Transmission System Operators for Electricity, ENTSO-E) . . . . . | 55        |
| <i>Vernetzte Geräte</i> . . . . .   | 54, 59    |



---

|  |     |
|--|-----|
| <i>VM-basiert</i> . . . . .  | 37  |
| <i>Web Services Description Language</i> (engl., WSDL) . . . . .                 | 38  |
| <i>Weiche Echtzeitanforderungen</i> . . . . .                                    | 167 |
| <i>Zeit-abhängige Tarifgestaltung</i> (engl. time of use pricing, TOU) . . . . . | 13  |
| <i>Zentralisierte Kommunikation</i> . . . . .                                    | 33  |
| <i>Zustandsidentifikation mehrerer Geräte</i> . . . . .                          | 120 |
| <i>Zustandsidentifikation von einzelnen Geräten</i> . . . . .                    | 120 |

## Symbolverzeichnis

|                       |   |          |
|-----------------------|---|----------|
| $\vec{a}$             | Ausgaben im Sinne der Gerätemodellierung . . . . .  | 65       |
| $A$                   | Anzahl der Ausprägungen des Betriebsstatus . . . . .  | 80       |
| $\delta$              | Zeitintervall zwischen 1 und der Mindest-Verbleib-Zeit $D_j$ . . . . .                                      | 121      |
| $D_j$                 | Mindest-Verbleib-Zeit . . . . .   | 121, 122 |
| $\vec{e}$             | Eingaben im Sinne der Gerätemodellierung . . . . .  | 65       |
| $G$                   | Anzahl der Geräte . . . . .   | 80       |
| $\text{Hist}\{P\}$    | Histogramm über die Leistungsaufnahme eines Geräts . . . . .  | 173      |
| $j$                   | Laufindex über die Anzahl der Entscheiderschwellen $J$ . . . . .  | 76, 117  |
| $J$                   | Anzahl der Entscheidungs-Intervalle . . . . .   | 76, 117  |
| $k$                   | Laufindex von 1 bis $K$ über die internen (Betriebs-)Zustände . . . . .                                     | 65, 79   |
| $K$                   | Anzahl der internen (Betriebs-)Zustände . . . . .   | 65, 79   |
| $l$                   | Laufindex von 1 bis $K$ über die internen (Betriebs-)Zustände . . . . .                                     | 79       |
| $\lambda$             | Vollständiger Regelsatz bestehend aus allen Entscheiderschwellen und ihren zugeordneten Zuständen . . . . . | 76       |
| $L_{\text{CPU}}$      | Nutzung der CPU angegeben zwischen 0 % und 100 % . . . . .  | 118      |
| $L_{\text{GPU}}$      | Nutzung der GPU angegeben zwischen 0 % und 100 % . . . . .  | 118      |
| $\mathcal{M}$         | Modell bestehend aus der Transformation $T_e$ und dem Regelsatz $\lambda$ . . . .                           | 76       |
| $M_{\text{Computer}}$ | Abbildung zur Ermittlung der geschätzten Leistungsaufnahme eines eingeschalteten Computers . . . . .        | 118      |
| $M_{\text{CPU}}$      | Abbildung der Nutzung der CPU auf den entsprechenden Leistungsanteil  | 118      |
| $M_{\text{GPU}}$      | Abbildung der Nutzung der GPU auf den entsprechenden Leistungsanteil  | 118      |
| $M_s$                 | Abbildung der internen Zustände $\vec{s}$ auf die Ausgaben $\vec{a}$ . . . . .                              | 66       |
| $m_v$                 | Element der Messdaten der Anzahl $V$ mit $v \in \{1, \dots, V\}$ . . . . .                                  | 66       |
| $N$                   | Anzahl der internen Zustandsausprägungen von $s$ . . . . .  | 75       |
| $N_k$                 | Anzahl der Ausprägungen des Zustands $s_k$ im Falle einer diskreten Modellierung . . . . .                  | 65       |
| $N_{\text{Pakete}}$   | Anzahl der übermittelten Pakete . . . . .   | 151      |

|                        |   |               |
|------------------------|---|---------------|
| $o_j$                  | Obere Schranke mit $o_j \in \mathbb{R}$ mit $j \in \{1, \dots, J\}$ , eindimensionale Modellierung . . . . .  | 76, 122       |
| $\vec{o}_j$            | Obere Schranke mit $o_{j,w} \in \mathbb{R}$ mit $j \in \{1, \dots, J\}$ und $w \in \{1, \dots, W\}$ , mehrdimensionale Modellierung . . . . .   | 76            |
| $P$                    | Leistung in Watt . . . . .  | 117, 119, 122 |
| $P_{\text{avg}}$       | Laufende Mittelwertbildung der gemessenen Leistung über die Intervallgröße $T_m$ . . . . .  | 122           |
| $P_{\text{Grundlast}}$ | Grundlastanteil an der Leistungsaufnahme eines Computers . . . . .  | 119           |
| $P_I, P_{II}$          | Näherungen für die Leistungsaufnahme eines Computers . . . . .  | 119           |
| $P_j$                  | Leistungsaufnahme in der Zustandsausprägung $S_j$ . . . . .   | 117           |
| $P_{\text{max}}$       | Maximale Leistungsaufnahme . . . . .  | 122           |
| $P_{\text{min}}$       | Minimale Leistungsaufnahme . . . . .  | 122           |
| $r$                    | Laufindex von 1 bis $R$ über die Elemente der Statusdaten $z_i$ . . . . .   | 66            |
| $\mathbb{R}$           | Menge der reellen Zahlen . . . . .  | 65, 76        |
| $R$                    | Anzahl der Statusdaten . . . . .  | 66            |
| $\vec{s}$              | Interne (Betriebs-)Zustände bestehend aus den Elementen $s_k \in \mathbb{S}_k$ der Anzahl $K$ mit $k \in \{1, \dots, K\}$ . . . . .   | 65, 79        |
| $s$                    | Interne Zustandsgröße eines Geräts als eindimensionaler, diskreter Betriebsstatus mit $N$ Ausprägungen $s \in \mathbb{S}^s = \{S_1, \dots, S_N\}$ . . . . .   | 75            |
| $\hat{s}$              | Geschätzter Zustand im Zustandsraum $\mathbb{S}^s$ . . . . .  | 76, 77        |
| $S$                    | eine Samplelänge . . . . .  | 122           |
| $\hat{s}_{\text{ANN}}$ | Geschätzter Zustand auf Basis von künstlichen neuronalen Netzen . . . . .   | 78            |
| $\hat{s}_{\text{HMM}}$ | Geschätzter Zustand auf Basis von Hidden-Markoff-Modellen . . . . .   | 77            |
| $S_j$                  | Einzelne diskrete Zustandsausprägung des Betriebsstatus $s$ im Zustandsraum $\mathbb{S}^s$ für die Zuordnung zu den Entscheidungsschwellen . . . . .  | 76, 117, 122  |
| $s_k$                  | Unabhängige interne Zustandsgröße der Anzahl $K$ mit $k \in \{1, \dots, K\}$ , zugehörig zu den internen (Betriebs-)Zuständen $\vec{s}$ , enthalten im Zustandsraum $\mathbb{S}_k$ der Anzahl $K$ mit $k \in \{1, \dots, K\}$ . . . . . | 65, 79        |
| $\hat{s}_k$            | Geschätzter Betriebszustand der Anzahl $K$ mit $k \in \{1, \dots, K\}$ . . . . .  | 79            |
| $\mathbb{S}_k$         | Zustandsraum des internen (Betriebs-)Zustands $s_k$ . . . . .   | 65, 79        |
| $S_{k,p}$              | Einzelne diskrete Zustandsausprägung des internen Zustands $s_k$ im Zustandsraum $\mathbb{S}_k$ . . . . .   | 65            |
| $S_n$                  | Einzelne diskrete Zustandsausprägung des Betriebsstatus $s$ im Zustandsraum $\mathbb{S}^s$ . . . . .  | 75            |
| $\mathbb{S}^s$         | Zustandsraum des eindimensionalen, diskreten Betriebsstatus $s$ . . . . .   | 75            |

|                           |   |         |
|---------------------------|---|---------|
| $\Delta t$                | Zeitdifferenz . . . . .   | 154     |
| $t$                       | Diskrete Zeitachse . . . . .  | 65      |
| $\Delta T$                | Abtastintervall . . . . .   | 65      |
| $T_{\text{ANN}}$          | Anwendung eines neuronalen Netzes . . . . .   | 78      |
| $\tau$                    | Gedächtnislänge . . . . .   | 66, 79  |
| $\delta\tau_{\text{RTT}}$ | Mittlere quadratische Abweichung der Paketumlaufzeit in Millisekunden,<br>auch als Jitter bezeichnet . . . . .                                    | 151     |
| $\tau_{\text{RTT}}$       | Mittlere Paketumlaufzeit in Millisekunden . . . . .   | 151     |
| $T_e$                     | Transformation der Sensordaten $\vec{e}$ z. B. im Sinne einer Merkmalsextraktion  | 76      |
| $\Delta t_{\text{Hop}}$   | Zusätzliche Latenz durch einen Hop . . . . .  | 154     |
| $T_m$                     | Intervallgröße der Mittelwertbildung . . . . .  | 122     |
| $\vec{u}_j$               | Untere Schranke mit $u_{j,w} \in \mathbb{R}$ mit $j \in \{1, \dots, J\}$ und $w \in \{1, \dots, W\}$ ,<br>mehrdimensionale Modellierung . . . . . | 76      |
| $u_j$                     | Untere Schranke mit $u_j \in \mathbb{R}$ mit $j \in \{1, \dots, J\}$ , eindimensionale Modellie-<br>rung . . . . .                                | 76, 122 |
| $v$                       | Laufindex von 1 bis $V$ über die Elemente der Messdaten $m_v$ . . . . .   | 66      |
| $V$                       | Anzahl der Messdaten . . . . .  | 66      |
| $w$                       | Laufindex von 1 bis $W$ über die mehrdimensionalen Beobachtungen $x_w$ . . . . .  | 76      |
| $W$                       | Anzahl der Dimensionen der Beobachtungen . . . . .  | 76      |
| $\vec{x}$                 | Mehrdimensionale Beobachtung bzw. Eingangswert des Entscheiders mit<br>den Elementen $x_w$ mit $w \in \{1, \dots, W\}$ . . . . .                  | 76, 77  |
| $x$                       | Eindimensionale Beobachtung bzw. Eingangswert des Entscheiders . . . . .  | 76      |
| $z_i$                     | Element der Statusdaten der Anzahl $R$ mit $r \in \{1, \dots, R\}$ . . . . .  | 66      |

# Abbildungsverzeichnis

|     |  |    |
|-----|--|----|
| 2.1 | Varianten des Demand-Side-Managements aufgetragen über die Zeitspanne des Eingriffs und über die Art der folgenden Energieverbrauchsreduzierung (nach [PD11, Bild 1]). Economic und Emergency DR stehen für zwei Varianten des Lastmanagements (engl. Demand Response). Die Art der Energieverbrauchsreduzierung beschreibt dabei, wie lange die Vorteile aus den unterschiedlichen Varianten gezogen werden können. Beim optimierten Ablauf werden die Vorteile aus vielen einzelnen Eingriffen gezogen. . . . .  | 12 |
| 2.2 | Maßnahmen des Energiemanagements (a) zur Reduktion, (b) zur Erhöhung und (c) zur Verschiebung des Energieeinsatzes (nach [KCM12, S. 5]). . . . .   | 18 |
| 2.3 | HEMS-Architektur mit Geräten in Hellblau, dezentralen Energieanlagen (DEA) in Lila sowie Energiespeichern in Dunkelblau. Durch eine Kommunikationsanbindung ist auch eine Beeinflussung durch externe Parteien möglich (in Anlehnung an [KCM12, Abbildungen 1 und 8], [PW16, Abbildung 1] und [Sha+18, Abbildung 1]) . . . . .   | 21 |
| 2.4 | Verallgemeinerter algorithmischer Ablauf eines HEMS, hier dargestellt in Form eines Reglers. Der Energieverbrauch und die Dienstqualität sind hier exemplarisch als Regelgrößen eingefügt. Viele Algorithmen arbeiten im Sinne einer Ablaufplanung (Scheduling). Üblicherweise handelt es sich bei den Regelgrößen um Pläne und Prognosen der Größen. (eigene Darstellung) . . .   | 24 |
| 3.1 | Schichtenmodell zur Auftrennung von Energiemanagementsystemen in verschiedene unabhängige Schichten. Durch die Konzeption nach diesem Schichtenmodell soll die Möglichkeit geschaffen werden, dass Funktionalitäten und Techniken verschiedener Schichten frei miteinander kombiniert werden können. Die untere Schicht bindet Geräte über verschiedene Techniken (Rechtecke) ein. Die Daten der Geräte werden über die mittlere Schicht der Kommunikation an die obere Schicht der Funktionalitäten vermittelt. Hier werden beispielhaft drei verschiedene Anwendungen zum Energiemanagement mit und ohne Nutzereinbindung implementiert. . . . . | 50 |

|      |   |    |
|------|---|----|
| 4.1  | Gerätemodell mit internen Zuständen $\vec{s}(t)$ , den Übergängen, die das Gerät charakterisieren, sowie Eingaben $\vec{e}(t)$ und Ausgaben $\vec{a}(t)$ . . . . .  | 65 |
| 4.2  | Übersicht – Wie kann der Betriebsstatus eines Gerätes ermittelt werden? . .   | 70 |
| 4.3  | (Regelbasiertes) Verarbeitungsschema zur Zustandsidentifikation anhand von Messdaten. Eine optionale Merkmalsextraktion erlaubt eine variable Vorverarbeitung der Messdaten. Die Parameter des Entscheiders werden als Modell zusammengefasst. . . . .  | 75 |
| 4.4  | Durch die Einbringung lokaler Koordinatoren (LK) ergibt sich gemeinsam mit den Geräten (G) eine hierarchische Struktur der Kommunikation. Die einzelnen Umgebungen, in die die lokalen Koordinatoren eingebracht wurden, sind mit gestrichelten Linien angedeutet. . . . .  | 82 |
| 4.5  | Struktur zur Visualisierung von Verbrauchsinformationen von Geräten. UI steht für Nutzerinterface (engl. User Interface), LK steht für die lokalen Koordinatoren und G für Geräte (vgl. Abschnitt 4.3). Die Messdaten werden in regelmäßigen Abständen an das Nutzerinterface gesendet. Initialisierungsbefehle sind nicht abgebildet, sie folgen aber den gleichen Pfaden wie die Messdaten. . . . . | 89 |
| 4.6  | Struktur zur Darstellung von Informationen zu einzelnen Gerätenutzungen. Die roten Quader symbolisieren eine Softwareverarbeitung. . . . .  | 90 |
| 4.7  | Navigationsstruktur für grafische Anzeigegeräte zur Anzeige von Energieverbrauchsdaten und -ereignissen . . . . .   | 90 |
| 4.8  | Struktur zur Steuerung auf Basis einer direkten Verknüpfung. Unter (a) wird eine direkte Geräteabstimmung gezeigt. Unter (b) wird eine Steuerung aufgrund eines Direkteingriffs im Sinne des Market Demand Response aus Unterabschnitt 2.1.2 realisiert. . . . .  | 92 |
| 4.9  | Struktur zur Steuerung auf Basis von Schwellenwerten. Diese kann für die Lastspitzenreduktion oder für die Steuerung auf Basis variabler Tarife verwendet werden. . . . .   | 92 |
| 4.10 | Struktur zur Abstimmung von Geräten (G) untereinander anhand von Optimierungsalgorithmen. Hier werden Prosumer dabei unterstützt, ihren Energieverbrauch und ihre lokale Energieerzeugung aufeinander abzustimmen. Die Wolken symbolisieren, dass sich hinter einem lokalen Koordinator noch weitere Koordinatoren und damit Ebenen befinden können. . . . .  | 94 |
| 4.11 | Struktur für Smart-Home-Anwendungen mit Visualisierung und Steuerung von Geräten über das Smartphone. Zusätzlich findet sich eine Szenensteuerung, die den Betriebsstatus mehrerer Geräte gleichzeitig einstellt. . . . .   | 95 |

|      |   |     |
|------|---|-----|
| 4.12 | Struktur für die prädiktive Instandhaltung anhand einer Überwachung der Messdaten von Geräten. Bei einem Fehler wird eine Alarmmeldung an die Verantwortliche bzw. den Verantwortlichen gesendet. . . . .   | 96  |
| 4.13 | Sicherheitsabschaltungen in Abhängigkeit einer Zeitdauer oder der Anwesenheit von Bewohnerinnen und Bewohner . . . . .  | 96  |
| 5.1  | Struktur der Implementierung unterteilt in Anwendungen, Netzwerkanbindung und IoT-Softwareplattform. Damit die Anwendungen Daten untereinander austauschen können, stellt die IoT-Softwareplattform den Anwendungen sogenannte Datenressourcen bereit. Auch ein Austausch über das Netzwerk ist mit Hilfe der Datenressourcen möglich. Folglich finden sich universelle Schnittstellen zu den Anwendungen, zu den Datenressourcen und für die Netzwerkanbindung. . . . .  | 98  |
| 5.2  | Initialisieren zweier Anwendungen durch die IoT-Softwareplattform. App 1 zeigt die Veröffentlichung einer Datenressource <b>node</b> . App 2 zeigt das Abonnieren einer fremden Datenressource <b>node</b> . Hierbei soll es sich um die gleiche Datenressource handeln. In Blau werden die Basismethoden dargestellt, die von den Anwendungen auf den Datenressourcen ausgeführt werden. In Rot sind die Initialisierungsaufrufe der IoT-Softwareplattform sowie die Weiterleitung der Datenveröffentlichungen durch die IoT-Softwareplattform (mit Hilfe von Callbacks) abgebildet. . . . . | 102 |
| 5.3  | Struktur bei abschließender Verarbeitung. Zunächst werden alle Messdaten gesammelt und nach Ende der Messung zusammen der Verarbeitung übergeben. . . . .   | 115 |
| 5.4  | Struktur bei andauernder direkter Verarbeitung. Die wiederkehrend aufgenommenen Messdaten werden in einer Datenressource veröffentlicht, die von den implementierten Erkennungs- und Nachbildungsverfahren in Form von Software-Anwendungen abonniert werden können. Die Ergebnisse werden in einer weiteren Datenressource veröffentlicht. . . . .   | 116 |
| 5.5  | Parameter des Modells. $S$ steht hier für eine Samplelänge. $T_m$ ist die Intervallgröße der Mittelwertbildung. (*) gilt nur bei $P_{\max} > 2,5$ Watt und $P_{\max} - P_{\min} > 3$ Watt. . . . .  | 122 |
| 5.6  | Ablauf des Abgleichs der Leistungsaufnahme mit verschiedenen Mustern, bestehend aus einer Merkmalsextraktion (ME), dem DTW-Musterabgleich und einer Auswahl des Musters mit der geringsten Abweichung. . . . .  | 124 |
| 5.7  | Parameter des Schwellenwert-Klassifikators, der dem DTW vorangestellt ist. $S$ steht hier wieder für eine Samplelänge. . . . .  | 125 |

- 5.8 Mess- und Demonstrationsaufbau zur Aufnahme von Netzspannungsbeeinflussungen (EMI). Mit der Auskopplung werden die auszuwertenden Signalanteile aus der Netzspannung gewonnen. Für die Erfassung der Messsignale der einzelnen Geräte entfernt der eingangsseitige Filter die Beeinflussungen anderer Geräte aus der Netzspannung, sodass in diesem Bereich nur noch die Beeinflussungen der lokalen Geräte vorliegen. . . . . 126
- 5.9 Struktur der Erkennung von mehreren Geräten anhand von künstlichen neuronalen Netzen (ANN). Die Struktur besteht zusätzlich aus einer Rahmenbildung und einer Merkmalsextraktion. Für die Merkmalsextraktion wird eine Fast-Fourier-Transformation verwendet. . . . . 127
- 5.10 Exemplarische Struktur an Datenressourcen. Repräsentiert werden zwei Räume mit insgesamt drei Geräten. Der Schreibtisch ist ein Unterbereich des Wohnzimmers. Räume und Geräte besitzen Datenressourcen für den Energieverbrauch. Die Geräte (grau hinterlegt) besitzen zusätzlich jeweils eine Datenressource, die den Status des Geräts beinhaltet. Für den Wasserkocher wurde zusätzlich eine weitere Analyse aktiviert, mit der die letzten Kochzeiten visualisiert werden können. Die Spalte Meta zeigt den ins Deutsche übersetzten Typ der Datenressource an, der in den Metadaten hinterlegt ist. Es werden die Grundtypen „Bereich“ für Navigationselemente, „Status“ für Datenobjekte und „Daten“ für Datenreihen unterschieden. Letztere sollen als Zeitreihen visualisiert werden können. . . . . 129
- 5.11 Die Display-Anwendung „Energiecockpit“ gibt einen Überblick über Geräte und ihre Energieverbräuche in einer Umgebung. Dargestellt werden zwei Maschinen mit Informationen zum Energieverbrauch, zu Temperaturen, zum aktuellen Betriebsstatus sowie zur Zeitdauer des letzten produktiven Betriebsstatus. . . . . 130
- 5.12 Die Smartphone-Anwendung „Meine Energie“ gibt der Nutzerin bzw. dem Nutzer direkte Rückmeldungen über ihre bzw. seine Gerätenutzung. Die linke Ansicht zeigt die Navigationsstruktur in Form einer Karte. Mittels Berührung eines Bereichs wird dieser dargestellt (rechte Ansicht). Dabei werden die Geräte in der Reihenfolge ihrer letzten Nutzung angezeigt. Neben der Nutzungszeit und -länge werden die jeweiligen Kosten der einzelnen Nutzungen angezeigt. Weiterhin können durch Auswahl eines einzelnen Gerätes die davorliegenden Nutzungen angezeigt werden. . . . . 132



- 5.13 Darstellung der Smartphone-Anwendung Home Control. Die linke Ansicht zeigt die Navigationsansicht. Geräte können hier direkt über Schaltflächen geschaltet werden. Die rechte Ansicht zeigt eine Detailseite, die z. B. durch Klick auf die Wertanzeige „Temperatur“ erreicht werden kann. . . . . 138
- 6.1 Für das Projekt EnEff Campus 2020 wurde ein Gesamtdemonstrator namens „Real-Life-Lab Campus 2020“ über alle Einzeldemonstratoren und erschlossenen Umgebungen realisiert. Das entstandene System überdeckt ausgewählte Umgebungen in verschiedenen Instituten der TU Braunschweig. Ebene 4 beinhaltet weitere Koordinatoren, die die Geräte in Ebene 5 zusammenfassen. 143
- 6.2 Prognose der Leistungsaufnahme zweier steuerbarer Kühltische (a) ohne Optimierung und (b) mit Optimierung. Abgebildet ist in Blau die Leistungsaufnahme des ersten Kühltisches sowie in Rot die Aufsummierung beider Leistungsaufnahmen. Ziel der Optimierung ist die möglichst genaue Ausnutzung einer dauerhaften Erzeugungsleistung von 120 W eines fiktiven lokalen Energieerzeugers bei möglichst geringem Energiebezug aus dem Versorgungsnetz. . . . . 147
- 6.3 Struktur des Evaluationsframeworks, bestehend aus einem Kontroll-Server (KS) mit Worker-Clients (W), einem Kommando-Client (K), der Aufgaben über den Kontrollserver an die Worker-Clients sendet und einem Test-Server (TS), der von den Worker-Clients angesprochen wird. Das Framework gestattet es, mittels vieler Worker-Clients den Test-Server anzusprechen und seine Leistungsfähigkeit zu protokollieren. . . . . 150
- 6.4 Darstellung der (a) mittleren Paketumlaufzeit und (b) deren mittlere quadratische Abweichung bei einem RPC-Aufruf zwischen zwei lokalen Koordinatoren aufgetragen gegen die Anzahl der übermittelten Pakete. . . . . 151
- 6.5 Mittlere Paketumlaufzeit zwischen zwei lokalen Koordinatoren unter Nutzung des Publish-Subscribe-Entwurfsmusters aufgetragen gegen die Anzahl der übermittelten Pakete. In Rot unter Nutzung des Protokolls aus Unterabschnitt 5.1.4, in Blau unter Nutzung des MQTT-Protokolls (siehe Unterabschnitt 2.3.4). (a) und (b) zeigen verschiedene Wertebereiche der gleichen Daten. . . 153
- 6.6 Darstellung der Datenraten in zwei Fällen: Unter (a) werden alle Geräte direkt mit einem zentralen Koordinator verbunden. Unter (b) werden zusätzlich lokale Koordinatoren zwischengeschaltet. Links sind jeweils die Anzahl der Entitäten dargestellt und rechts die Eingangs- bzw. Ausgangsdatenraten. . . 156
- 6.7 Darstellung des Messzyklus ansteigend von 0 bis 100 % CPU-Last und direkt zurück auf 0 % CPU-Last in sekundlicher Auflösung der Computer A-1 (oben) und D-1 (unten). . . . . 171

|      |  |     |
|------|--|-----|
| 6.8  | Abweichung von geschätztem zu tatsächlichem Energieverbrauch pro Computer. Die Standardabweichung in Rot wurde über die Einzelwerte ermittelt. Die normierte Gesamtabweichung in Blau wurde über die gesamte Datenreihe akkumuliert. Für die Ermittlung wurde eine lineare Schätzung verwendet.  | 172 |
| 6.9  | Ausschnitt des zeitlichen Verlaufs der Leistungsaufnahme eines Wasserkochers mit zwei Aktivitäten (links) sowie Histogramm der Leistungsaufnahme über 17 Tage (rechts). In Grau ist eine exemplarische Entscheiderschwelle bei 1.100 Watt eingezeichnet.   | 173 |
| 6.10 | Darstellung der Anzahlen der Geräten, deren jeweils momentaner Betriebsstatus vollständig korrekt erkannt werden konnte, im Vergleich zu den verbleibenden Geräte unter Verwendung des Modells aus Unterabschnitt 5.2.7 aufgeteilt nach verschiedenen Kategorien: Links – alle Geräte, Mitte – Aufteilung der Geräte nach Aufgaben, Rechts – Aufteilung in Nicht-Programmgesteuerte und Programm-gesteuerte Geräte.  | 174 |
| 6.11 | Ausschnitt des zeitlichen Verlaufs der Leistungsaufnahme einer Waschmaschine (links). Dargestellt ist ein einzelnes Waschprogramm. Zusätzlich ist die Leistungsaufnahme über 10 Tage als Histogramm abgebildet (rechts). Die in Grau dargestellte exemplarische Entscheiderschwelle bei 1.000 Watt bildet den Betriebsstatus nicht korrekt ab.   | 175 |
| 6.12 | Frequenz-basierte Merkmale eines USB-Netzteils im Verlauf über 0,25 Sekunden. Die Darstellung der Helligkeitsskala wurde logarithmiert. Die Skala wurde auf den Eingangspegel zurückgerechnet.   | 177 |
| 6.13 | Darstellung der Fehlerrate bei der Zuordnung von einzelnen Geräten und verschiedene Kombinationen von Geräten sowie die Gesamtfehlerrate über alle Kombinationen.  | 178 |
| 6.14 | Darstellung der Erkennung aller Geräte gleichzeitig im Vergleich zur Erkennung aller Geräte ohne den PC-Monitor. Aufgetragen sind die Modellausgabewerte des künstlichen neuronalen Netzes der zugeordneten Geräte. Der dargestellte Wert wurde jeweils aus 512 Rahmen gemittelt. Auffällig ist die Änderung des Erkennungswerts in Abhängigkeit der Aktivität des Monitors beim hier betrachteten USB-Ladegerät. Dabei ist festzustellen, dass die Erkennung aller Geräte ohne eingesteckten PC-Monitor eine sehr gute Fehlerrate aufweist. | 179 |

## Verzeichnis der eigenen Veröffentlichungen

- [1] David Scheler, Simon Walz und Tim Fingscheidt. „On Iterative Exchange of Soft State Information in Two-Channel Automatic Speech Recognition“. In Proc. *Speech Communication; 10. ITG Symposium*, Nr. 1.3, S. 26–29, 2012, URL: <https://ieeexplore.ieee.org/abstract/document/6309578>.
- [2] Simon Walz. „Internet of Things: Visualisierung des Energieverbrauchs im IoT“. 27. *FKTG-Fachtagung 2016*, Leipzig, 10. Mai 2016.
- [3] Simon Walz und Yannic Schröder. „A privacy-preserving system architecture for applications raising the energy efficiency“. In Proc. *6th IEEE International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, elektronisch (5 Seiten), Berlin, September 2016, DOI: 10.1109/ICCE-Berlin.2016.7684718.
- [4] Simon Walz. „Energie sparen mit Hilfe des Internet der Dinge (IoT)“. *FKT Fachzeitschrift für Fernsehen, Film und elektronische Medien*, Schiele & Schön, Nr. 8-9, S. 366–370, 2016.
- [5] Simon Walz. „Eine Softwareplattform zur Realisierung von verteilten Smart-Energy-Anwendungen im IoT“. In Proc. *VDE-Kongress 2016 – Internet der Dinge*, elektronisch (5 Seiten), Mannheim, November 2016, ISBN: 978-3-8007-4308-7.
- [6] Simon Walz und Yannic Schröder. „Workshop Real-Life-Lab-Campus TUBS – Transparenz und intelligente Steuerung des Energieverbrauchs“. *EnEff Campus 2020 – Umsetzung des integralen Masterplans zur energetischen Optimierung des Campus an der TU Braunschweig*, Braunschweig, 9. November 2016.

## Verzeichnis der betreuten studentischen Arbeiten

- [13/007] *Masterarbeit:* Hurtig, Thomas: Entwurf einer Energieberatung per Smartphone
- [13/008] *Masterarbeit:* Franzky, Jonathan: Implementierung eines Energie-Geräteerkenners
- [13/703] *Medientechnisches Projekt:* Fritzsche, Sven; Richter, Maj Kristin: Steuerung einer Multimediaanlage mit energietechnischen Zusatzfunktionen
- [13/710] *Bachelorarbeit:* Filusch, David: Erarbeitung von Konzepten zur intelligenten Steuerung der Beleuchtung von Räumen und Fluren
- [14/003] *Masterarbeit:* Maurer, Lorenz: Entwicklung einer Smartphone-Applikation zur Energiekosteneinsparung: Nutzereinbindung und Visualisierungskonzepte
- [14/013] *Masterarbeit:* Sarodnick, Julia: Vergleich von Einsatzmöglichkeiten von Augmented Reality-basierten Visualisierungsverfahren mittels Realisierung eines 3D-objektbasierten Trackingverfahrens in einer Lernanwendung
- [14/702] *Bachelorarbeit:* Schwarz, Timo: Konzepte zur intelligenten Steuerung der Energiesparfunktionen Windows-basierter Systeme
- [14/704] *Bachelorarbeit:* Knibbe, Christoph: Detektion von Rückwirkungen elektrischer Geräte auf die Netzspannung des Niederspannungsnetzes
- [15/002] *Masterarbeit:* Buchner, Mira: Entwurf einer Tablet-Anwendung zur Aufschlüsselung des Energieverbrauchs privater Haushalte
- [15/702] *Bachelorarbeit:* Pollack, Alexander: Entwurf und Implementierung einer modularen Merkmalsextraktion für die Energie-Geräte-Erkennung
- [15/704] *Bachelorarbeit:* Dell, Paul: Schätzung des Energieverbrauchs von Desktop-Computern
- [15/800] *Projektarbeit:* Kloppenburg, Beate: Equipmentverwaltungs- und Ausleihsystem via Barcodescanner und MySQL

- 
- [16/003] *Masterarbeit:* Taubner, Anne: Energieverbrauchs-Informationen in Echtzeit per Smartphone-Anwendung
- [16/011] *Masterarbeit:* Fromme, Harro Fabian: Entwicklung eines webbasierten Energieleitstandes für den industriellen Einsatz
- [16/014] *Masterarbeit:* Suo, Chao: Entwicklung einer verteilten IoT-Anwendungsplattform
- [16/015] *Masterarbeit:* Kniep, Christian: Erkennung von elektrischen Geräten anhand hochfrequenter Störungen der Netzspannung
- [16/019] *Masterarbeit:* Bellekom, Saskia: Datensicherheit in einer universitätsweiten Anwendung zur Erhöhung der Energieeffizienz
- [17/017] *Masterarbeit:* Schubert, Marcel: Ein Privatsphäre-schützendes Energiemanagementsystem

## Verzeichnis der referenzierten Gesetze

|              |   |
|--------------|---|
| AbLaV        | Verordnung zu abschaltbaren Lasten vom 16. August 2016 (BGBl. I S. 1984), die zuletzt durch Artikel 9 des Gesetzes vom 22. Dezember 2016 (BGBl. I S. 3106) geändert worden ist.                       |
| BDSG         | Bundesdatenschutzgesetz vom 30. Juni 2017 (BGBl. I S. 2097).  |
| EEG 2017     | Erneuerbare-Energien-Gesetz vom 21. Juli 2014 (BGBl. I S. 1066), das zuletzt durch Artikel 1 des Gesetzes vom 21. Juni 2018 (BGBl. I S. 862) geändert worden ist.                                     |
| EnEG         | Energieeinsparungsgesetz in der Fassung der Bekanntmachung vom 1. September 2005 (BGBl. I S. 2684), das zuletzt durch Artikel 1 des Gesetzes vom 4. Juli 2013 (BGBl. I S. 2197) geändert worden ist.  |
| EnVKG        | Energieverbrauchskennzeichnungsgesetz vom 10. Mai 2012 (BGBl. I S. 1070), das zuletzt durch Artikel 1 des Gesetzes vom 10. Dezember 2015 (BGBl. I S. 2194) geändert worden ist.                       |
| EnWG         | Energiewirtschaftsgesetz vom 7. Juli 2005 (BGBl. I S. 1970, 3621), das zuletzt durch Artikel 3 des Gesetzes vom 17. Dezember 2018 (BGBl. I S. 2549) geändert worden ist.                              |
| EU 2017/1369 | Verordnung (EU) 2017/1369 des Europäischen Parlaments und des Rates vom 4. Juli 2017 zur Festlegung eines Rahmens für die Energieverbrauchskennzeichnung und zur Aufhebung der Richtlinie 2010/30/EU. |
| GeschGehG    | Gesetz zum Schutz von Geschäftsgeheimnissen vom 18. April 2019 (BGBl. I S. 466).  |
| KWKG         | Kraft-Wärme-Kopplungsgesetz vom 21. Dezember 2015 (BGBl. I S. 2498), das zuletzt durch Artikel 6 des Gesetzes vom 13. Mai 2019 (BGBl. I S. 706) geändert worden ist.                                  |
| StromNev     | Stromnetzentgeltverordnung vom 25. Juli 2005 (BGBl. I S. 2225), die zuletzt durch Artikel 10 des Gesetzes vom 13. Mai 2019 (BGBl. I S. 706) geändert worden ist.                                      |

---

|          |  |
|----------|--|
| StromStG | Stromsteuergesetz vom 24. März 1999 (BGBl. I S. 378; 2000 I S. 147), das zuletzt durch Artikel 4 des Gesetzes vom 27. August 2017 (BGBl. I S. 3299; 2018 I 126) geändert worden ist. |
| MsbG     | Messstellenbetriebsgesetz vom 29. August 2016 (BGBl. I S. 2034), das durch Artikel 15 des Gesetzes vom 22. Dezember 2016 (BGBl. I S. 3106) geändert worden ist.                      |

## Literaturverzeichnis

- [AB18] Umar Ahsan und Abdul Bais. „Distributed Smart Home Architecture for Data Handling in Smart Grid“. In: *Canadian Journal of Electrical and Computer Engineering* 41.1 (2018), Seiten 17–27. ISSN: 08408688. DOI: 10.1109/CJECE.2017.2776975.
- [Abb+18] Nasir Abbas, Yan Zhang, Amir Taherkordi und Tor Skeie. „Mobile Edge Computing: A Survey“. In: *IEEE Internet of Things Journal* 5.1 (Feb. 2018), Seiten 450–465. ISSN: 2327-4662. DOI: 10.1109/JIOT.2017.2750180.
- [AE07] M. H. Albadi und E. F. El-Saadany. „Demand Response in Electricity Markets: An Overview“. In: *2007 IEEE Power Engineering Society General Meeting*. IEEE, Juni 2007, Seiten 1–5. ISBN: 1-4244-1296-X. DOI: 10.1109/PES.2007.385728.
- [AE12] Acatech und EIT ICT Labs. *Future Energy Grid*. Herausgegeben von Hans-Jürgen Appelrath, Henning Kagermann und Christoph Mayer. acatech STUDIE. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, Seite 325. ISBN: 978-3-642-27863-1. DOI: 10.1007/978-3-642-27864-8.
- [Amp+16] Amprion, 50Hertz, Tennet und Transnet BW. *Leitfaden zur Präqualifikation von Windenergieanlagen zur Erbringung von Minutenreserveleistung im Rahmen einer Pilotphase*. Technischer Bericht. 2016, Seiten 1–8. URL: <https://www.regelleistung.net/>.
- [Amp+18] Amprion, 50Hertz, Tennet und Transnet BW. *Präqualifikationsverfahren für Regelreserveanbieter in Deutschland*. Technischer Bericht November. 2018. URL: <https://www.regelleistung.net/>.
- [Ant+18] Francesco Antonini, Antonio Giannitrapani, Simone Paoletti und Antonio Vicino. „Optimal management of energy storage systems for residential customers with photovoltaic generation“. In: *2018 IEEE International Energy Conference (ENERGYCON)*. IEEE, Juni 2018, Seiten 1–6. ISBN: 978-1-5386-3669-5. DOI: 10.1109/ENERGYCON.2018.8398806.



- 
- [AS18] S. L. Arun und M. P. Selvan. „Intelligent Residential Energy Management System for Dynamic Demand Response in Smart Buildings“. In: *IEEE Systems Journal* 12.2 (2018), Seiten 1329–1340. ISSN: 19379234. DOI: 10.1109/JSYST.2017.2647759.
- [AV16] Mohammad Abdullah Al Faruque und Korosh Vatanparvar. „Energy Management-as-a-Service Over Fog Computing Platform“. In: *IEEE Internet of Things Journal* 3.2 (Apr. 2016), Seiten 161–169. ISSN: 2327-4662. DOI: 10.1109/JIOT.2015.2471260.
- [Awa+18] Muhammad Awais, Muhammad Babar Rasheed, Muhammad Zahid Iqbal und Muhammad Usman. „Multiple user based residential energy management scheme for smart homes“. In: *Proceedings - 2018, IEEE 1st International Conference on Power, Energy and Smart Grid, ICPESG 2018* (2018), Seiten 1–6. DOI: 10.1109/ICPESG.2018.8384492.
- [AZH18] Mohammad Aazam, Sherali Zeadally und Khaled A. Harras. „Fog Computing Architecture, Evaluation, and Future Research Directions“. In: *IEEE Communications Magazine* 56.5 (Mai 2018), Seiten 46–52. ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1700707.
- [Bar15] Lyn Bartram. „Design Challenges and Opportunities for Eco-Feedback in the Home“. In: *IEEE Computer Graphics and Applications* 35.4 (Juli 2015), Seiten 52–62. ISSN: 0272-1716. DOI: 10.1109/MCG.2015.69.
- [Ber02] T. Berry. „Standards for energy management system application program interfaces“. In: *DRPT2000. International Conference on Electric Utility Deregulation and Restructuring and Power Technologies. Proceedings (Cat. No.00EX382)*. Cim. IEEE, 2002, Seiten 156–161. ISBN: 0-7803-5902-X. DOI: 10.1109/DRPT.2000.855656.
- [BF17] Oliver Bates und Adrian Friday. „Beyond Data in the Smart City: Repurposing Existing Campus IoT“. In: *IEEE Pervasive Computing* 16.2 (2017), Seiten 54–60. ISSN: 1536-1268. DOI: 10.1109/MPRV.2017.30.
- [BG14] Andrew Banks und Rahul Gupta, Herausgeber. *MQTT Version 3.1.1- Specification*. 2014. URL: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/os/mqtt-v3.1.1-os.doc>.
- [BHV14] Thomas Bauernhansl, Michael ten Hompel und Birgit Vogel-Heuser, Herausgeber. *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014. ISBN: 978-3-658-04681-1. DOI: 10.1007/978-3-658-04682-8.

- [Bi+18] Yuanguo Bi, Guangjie Han, Chuan Lin u. a. „Mobility Support for Fog Computing: An SDN Approach“. In: *IEEE Communications Magazine* 56.5 (Mai 2018), Seiten 53–59. ISSN: 0163-6804. DOI: 10.1109/MCOM.2018.1700908.
- [BL14] Michael Blackstock und Rodger Lea. „IoT interoperability: A hub-based approach“. In: *2014 International Conference on the Internet of Things (IOT)*. Band 22. 3. IEEE, Okt. 2014, Seiten 79–84. ISBN: 978-1-4799-5154-3. DOI: 10.1109/IOT.2014.7030119.
- [BMW17] BMWi. „Eckpunktepapier Mieterstrom“. In: *BMWi* (2017), Seiten 1–5. URL: <https://www.bmwi.de/Redaktion/DE/Downloads/E/eckpunkte-mieterstrom.pdf>.
- [Bra17] Tim Bray. *The JavaScript Object Notation (JSON) Data Interchange Format*. RFC 8259 (Proposed Standard). 2017. URL: <http://www.ietf.org/rfc/rfc7159.txt>.
- [Bun10] Bundesregierung. *Energiekonzept für eine umweltschonende, zuverlässige und bezahlbare Energieversorgung*. Technischer Bericht. 2010, Seiten 1–40.
- [Bun13] Bundesregierung. *Die Energiewende kommt voran*. Technischer Bericht. 2013, Seiten 1–11.
- [Bun16] Bundesamt für Wirtschaft und Ausfuhrkontrolle. *Energiemanagementsysteme - Liste förderfähiger Energiemanagementsoftware*. Technischer Bericht. Eschborn, 2016, Seite 11.
- [Bun18] Bundesministerium für Wirtschaft und Energie. *Erneuerbare Energien in Zahlen - Nationale und internationale Entwicklung im Jahr 2017*. Technischer Bericht. 2018, Seite 80. URL: <http://www.erneuerbare-energien.de>.
- [BZ15] Marc Beaudin und Hamidreza Zareipour. „Home energy management systems: A review of modelling and complexity“. In: *Renewable and Sustainable Energy Reviews* 45 (2015), Seiten 318–335. ISSN: 13640321. DOI: 10.1016/j.rser.2015.01.046.
- [BZ16] Oladayo Bello und Sherali Zeadally. „Intelligent Device-to-Device Communication in the Internet of Things“. In: *IEEE Systems Journal* 10.3 (Sep. 2016), Seiten 1172–1182. ISSN: 1932-8184. DOI: 10.1109/JSYST.2014.2298837.
- [CC18] Lokesh Chandra und Saurabh Chanana. „Energy Management of Smart Homes with Energy Storage, Rooftop PV and Electric Vehicle“. In: *2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS)*. IEEE, Feb. 2018, Seiten 1–6. ISBN: 978-1-5386-2663-4. DOI: 10.1109/SCEECS.2018.8546857.

- 
- [CD17] Tanmay Chakraborty und Soumya Kanti Datta. „Home automation using edge computing and Internet of Things“. In: *2017 IEEE International Symposium on Consumer Electronics (ISCE)*. IEEE, Nov. 2017, Seiten 47–49. ISBN: 978-1-5386-2189-9. DOI: 10.1109/ISCE.2017.8355544.
- [Cel+18] Berk Celik, Robin Roche, David Bouquain und Abdellatif Miraoui. „Decentralized Neighborhood Energy Management With Coordinated Smart Home Energy Sharing“. In: *IEEE Transactions on Smart Grid* 9.6 (Nov. 2018), Seiten 6387–6397. ISSN: 1949-3053. DOI: 10.1109/TSG.2017.2710358.
- [Cha+19] Hsuan-Hao Chang, Wei-Yu Chiu, Hongjian Sun und Chia-Ming Chen. „User-Centric Multiobjective Approach to Privacy Preservation and Energy Cost Minimization in Smart Home“. In: *IEEE Systems Journal* 13.1 (März 2019), Seiten 1030–1041. ISSN: 1932-8184. DOI: 10.1109/JSYST.2018.2876345.
- [Che+18] Nanxi Chen, Yang Yang, Tao Zhang u. a. „Fog as a Service Technology“. In: *IEEE Communications Magazine* 56.11 (2018), Seiten 95–101. ISSN: 15581896. DOI: 10.1109/MCOM.2017.1700465.
- [CK13] Stuart Cheshire und Marc Krochmal. *DNS-based service discovery*. RFC 6763 (Proposed Standard). 2013. URL: <http://www.ietf.org/rfc/rfc6763.txt>.
- [Con+12] Davide Conzon, Thomas Bolognesi, Paolo Brizzi u. a. „The VIRTUS Middleware: An XMPP Based Architecture for Secure IoT Communications“. In: *2012 21st International Conference on Computer Communications and Networks (ICCCN)*. IEEE, Juli 2012, Seiten 1–6. ISBN: 978-1-4673-1544-9. DOI: 10.1109/ICCCN.2012.6289309.
- [Dao+18] Van Tu Dao, Hideo Ishii, Yasuhiro Hayashi u. a. „Optimal Responses of Home Energy Management Systems to Aggregator and Utility Requirements“. In: *2018 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*. IEEE, Mai 2018, Seiten 616–621. ISBN: 978-1-5386-4291-7. DOI: 10.1109/ISGT-Asia.2018.8467883.
- [Dar18] Sarah J. Darby. „Smart technology in the home: time for more clarity“. In: *Building Research & Information* 46.1 (Jan. 2018), Seiten 140–147. ISSN: 0961-3218. DOI: 10.1080/09613218.2017.1301707.
- [DB16] Soumya Kanti Datta und Christian Bonnet. „Describing things in the Internet of Things: From CoRE link format to semantic based descriptions“. In: *2016 IEEE International Conference on Consumer Electronics-Taiwan, ICCE-TW 2016* i (2016), Seiten 1–2. ISSN: 03029743. DOI: 10.1109/ICCE-TW.2016.7520965.

- [Din+13] Hoang T. Dinh, Chonho Lee, Dusit Niyato und Ping Wang. „A survey of mobile cloud computing: architecture, applications, and approaches“. In: *Wireless Communications and Mobile Computing* 13.18 (Dez. 2013), Seiten 1587–1611. ISSN: 15308669. DOI: 10.1002/wcm.1203.
- [Dol+18] Valeria Dolce, Courtney Jackson, Simone Silvestri u. a. „Social-Behavioral Aware Optimization of Energy Consumption in Smart Homes“. In: *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, Juni 2018, Seiten 163–172. ISBN: 978-1-5386-5470-5. DOI: 10.1109/DCOSS.2018.00033.
- [Ebr+18] Mahoor Ebrahimi, Amin Shokri Gzafroudi, Juan Manuel Corchado und Mahan Ebrahimi. „Energy Management of Smart Home Considering Residences’ Satisfaction and PHEV“. In: *2018 International Conference on Smart Energy Systems and Technologies (SEST)*. IEEE, Sep. 2018, Seiten 1–6. ISBN: 978-1-5386-5326-5. DOI: 10.1109/SEST.2018.8495707.
- [Eja+17] Waleed Ejaz, Muhammad Naeem, Adnan Shahid u. a. „Efficient Energy Management for the Internet of Things in Smart Cities“. In: *IEEE Communications Magazine* 55.1 (Jan. 2017), Seiten 84–91. ISSN: 0163-6804. DOI: 10.1109/MCOM.2017.1600218CM.
- [Ell14] Eric Elliott. *Programming JavaScript Applications*. 1. 2014, Seiten 1–5. ISBN: 9780874216561. DOI: 10.1007/s13398-014-0173-7.2.
- [ENT18] ENTSO-E. *ELECTRICITY IN EUROPE 2017 Synthetic – Synthetic overview of electric system consumption, generation and exchanges in 34 European countries*. Technischer Bericht. 2018, Seite 20.
- [Err+18] Sai Mounika Errapotu, Jingyi Wang, Yanmin Gong u. a. „SAFE: Secure Appliance Scheduling for Flexible and Efficient Energy Consumption for Smart Home IoT“. In: *IEEE Internet of Things Journal* 5.6 (Dez. 2018), Seiten 4380–4391. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2866998.
- [Eug+03] Patrick TH. Eugster, Pascal A. Felber, Rachid Guerraoui und Anne-Marie Kermarrec. „The many faces of publish/subscribe“. In: *ACM Computing Surveys* 35.2 (2003), Seiten 114–131. ISSN: 03600300. DOI: 10.1145/857076.857078.
- [Eur12] Europäisches Parlament und Europäischer Rat. „Richtlinie 2012/27/EU zur Energieeffizienz“. In: *Amtsblatt der Europäischen Union* November 2010 (2012), Seiten 1–56. URL: <http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2012:315:0001:0056:DE:PDF>.

- 
- [Eur14] Europäischer Rat. *Schlussfolgerungen zum Rahmen für die Klima- und Energiepolitik bis 2030*. 2014.
- [Eur18a] Europäisches Parlament. *Legislative Entschließung des Europäischen Parlaments vom 13. November 2018 zu dem Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates zur Änderung der Richtlinie 2012/27/EU zur Energieeffizienz*. 2018.
- [Eur18b] Europäisches Parlament. *Legislative Entschließung des Europäischen Parlaments vom 13. November 2018 zu dem Vorschlag für eine Richtlinie des Europäischen Parlaments und des Rates zur Förderung der Nutzung von Energie aus erneuerbaren Quellen*. 2018.
- [FM11] Ian Fette und Alexey Melnikov. „The WebSocket Protocol“. In: Request for Comments 6455 (2011), Seiten 1–71. URL: <http://www.ietf.org/rfc/rfc6455.txt>.
- [Fro+11] Jon Froehlich, Eric Larson, Sidhant Gupta u. a. „Disaggregated End-Use Energy Sensing for the Smart Grid“. In: *IEEE Pervasive Computing* 10.1 (Jan. 2011), Seiten 28–39. ISSN: 1536-1268. DOI: 10.1109/MPRV.2010.74.
- [GOZ08] Veronika Grimm, Axel Ockenfels und Gregor Zoetl. „Strommarktdesign: Zur Ausgestaltung der Auktionsregeln an der EEX“. In: *Zeitschrift für Energiewirtschaft* 32.3 (2008), Seiten 147–161. ISSN: 0343-5377. DOI: 10.1007/s12398-008-0020-7.
- [Han+18] Timothy M. Hansen, Edwin K.P. Chong, Siddharth Suryanarayanan u. a. „A partially observable markov decision process approach to residential home energy management“. In: *IEEE Transactions on Smart Grid* 9.2 (2018), Seiten 1271–1281. ISSN: 19493053. DOI: 10.1109/TSG.2016.2582701.
- [Has+18] Najmul Hassan, Saira Gillani, Ejaz Ahmed u. a. „The Role of Edge Computing in Internet of Things“. In: *IEEE Communications Magazine* 56.11 (2018), Seiten 110–115. ISSN: 15581896. DOI: 10.1109/MCOM.2018.1700906.
- [Hei16] Daniel Heidorn. „Lokale bedarfsangepasste Steuerung für elektrische Verbraucher am Beispiel einer Kühlschrankssteuerung“. Bachelorarbeit. TU Braunschweig. Institut für Betriebssysteme und Rechnerverbund, 2016. URL: <https://www.ibr.cs.tu-bs.de/theses/buesch/enneffridge.html>.
- [JB04] Dirk Jansen und Holger Buttner. „Real-time Ethernet: the EtherCAT solution“. In: *Computing and Control Engineering* 15.1 (2004), Seiten 16–21.
- [Jen17] Scott Jenson. „The Future IoT: Building Better Legos“. In: *Computer* 50.2 (Feb. 2017), Seiten 68–71. ISSN: 0018-9162. DOI: 10.1109/MC.2017.48.

- [JG11] Wayne Jansen und Timothy Grance. *Guidelines on security and privacy in public cloud computing*. Technischer Bericht. Gaithersburg, MD: National Institute of Standards und Technology, 2011. DOI: 10.6028/NIST.SP.800-144.
- [Kah+12] Walter Kahlenborn, Sibylle Kabisch, Johanna Klein und Ina Richter. *Energiemanagementsysteme in der Praxis – ISO 50001: Leitfaden für Unternehmen und Organisationen*. Technischer Bericht. Berlin: Bundesministerium für Umwelt, Naturschutz und Reaktorsicherheit (BMU), 2012, Seite 113. URL: <https://www.umweltbundesamt.de/sites/default/files/medien/publikation/long/3959.pdf>.
- [KCM12] Aravind Kailas, Valentina Cecchi und Arindam Mukherjee. „A survey of communications and networking technologies for energy management in buildings and home automation“. In: *Journal of Computer Networks and Communications* 2012 (2012). ISSN: 20907141. DOI: 10.1155/2012/932181.
- [KJ11] J Zico Kolter und Matthew J Johnson. „REDD : A Public Data Set for Energy Disaggregation Research“. In: *Workshop on Data Mining Applications in Sustainability (SIGKDD)*. 2011, Seiten 1–6. ISBN: 9781450308403.
- [KK16] Jack Kelly und William Knottenbelt. „Does disaggregated electricity feedback reduce domestic electricity consumption? A systematic review of the literature“. In: *TEDDINET Energy Feedback Symposium 2016* (2016), Seiten 65–73. arXiv: 1605.00962.
- [KKA19] Sarmadullah Khan, Rafiullah Khan und Ali Hilal Al-Bayatti. „Secure Communication Architecture for Dynamic Energy Management in Smart Grid“. In: *IEEE Power and Energy Technology Systems Journal* 6.1 (März 2019), Seiten 47–58. ISSN: 2332-7707. DOI: 10.1109/JPETS.2019.2891509.
- [Klo+06] Marian Klobasa, Thomas Erge, Aleksandra Sasa Bukvic-Schäfer und Maik Hollmann. „Demand Side Management in dezentral geführten Verteilnetzen (Erfahrungen und Perspektiven)“. In: *11th Kasseler Symposium Energy Systems Technology*. Kassel, 2006, Seiten 115–134.
- [Kri+18] Prakash N. Krishna, Suraj R. Gupta, P.V. Shankaranarayanan u. a. „Fuzzy Logic Based Smart Home Energy Management System“. In: *2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*. IEEE, Juli 2018, Seiten 1–5. ISBN: 978-1-5386-4430-0. DOI: 10.1109/ICCCNT.2018.8493744.

- 
- [KSG18] Zivko Kokolanski, Tomislav Shuminoski und Cvetan Gavrovski. „Architectures and Challenges for the Household Energy Management Systems“. In: *2018 IEEE XXVII International Scientific Conference Electronics - ET*. IEEE, Sep. 2018, Seiten 1–4. ISBN: 978-1-5386-6692-0. DOI: 10.1109/ET.2018.8549610.
- [KTH13] Dominik Klein, Phuoc Tran-Gia und Matthias Hartmann. „Big Data“. In: *Informatik-Spektrum* 36.3 (Juni 2013), Seiten 319–323. ISSN: 0170-6012. DOI: 10.1007/s00287-013-0702-3.
- [KVC18] Chanaka Keerthisinghe, Gregor Verbic und Archie C. Chapman. „A Fast Technique for Smart Home Management: ADP With Temporal Difference Learning“. In: *IEEE Transactions on Smart Grid* 9.4 (Juli 2018), Seiten 3291–3303. ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2629470.
- [Lee+17] Ying-Tsung Lee, Wei-Hsuan Hsiao, Yan-Shao Lin und Seng-Cho T. Chou. „Privacy-preserving data analytics in cloud-based smart home with community hierarchy“. In: *IEEE Transactions on Consumer Electronics* 63.2 (Mai 2017), Seiten 200–207. ISSN: 0098-3063. DOI: 10.1109/TCE.2017.014777.
- [Leh10] Sebastian Lehnhoff. *Dezentrales vernetztes Energiemanagement*. Dortmund: Vieweg+Teubner Verlag, 2010, Seite 264. ISBN: 9783834812704.
- [LGT08] Adam Lewis, Soumik Ghosh und Nian Feng. Tzeng. „Run-time Energy Consumption Estimation Based on Workload in Server Systems“. In: *Proceedings of the Workshop on Power Aware Computing and Systems (HotPower '08)* (2008), Seiten 17–21. URL: <https://www.usenix.org/legacy/event/hotpower08/tech/>.
- [Li+18a] Shenglin Li, Junjie Yang, Wen Zhan Song und An Chen. „A Real-Time Electricity Scheduling for Residential Home Energy Management“. In: *IEEE Internet of Things Journal* 4662.c (2018), Seiten 1–1. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2872463.
- [Li+18b] Weixian Li, Thillainathan Logenthiran, Van-Tung Phan und Wai Lok Woo. „Implemented IoT-Based Self-Learning Home Management System (SHMS) for Singapore“. In: *IEEE Internet of Things Journal* 5.3 (Juni 2018), Seiten 2212–2219. ISSN: 2327-4662. DOI: 10.1109/JIOT.2018.2828144.
- [Lin10] David Lindley. „Smart grids: The energy storage problem“. In: *Nature* 463.7277 (Jan. 2010), Seiten 18–20. ISSN: 0028-0836. DOI: 10.1038/463018a.

- [LL16] Chao-Hsien Lee und Ying Hsun Lai. „Design and implementation of a universal smart energy management gateway based on the Internet of Things platform“. In: *2016 IEEE International Conference on Consumer Electronics (ICCE)*. 1. IEEE, Jan. 2016, Seiten 67–68. ISBN: 978-1-4673-8364-6. DOI: 10.1109/ICCE.2016.7430524.
- [LLL11] Fengjun Li, Bo Luo und Peng Liu. „Secure and privacy-preserving information aggregation for smart grids“. In: *International Journal of Security and Networks* 6.1 (2011), Seite 28. ISSN: 1747-8405. DOI: 10.1504/IJSN.2011.039631.
- [Luo+19] Fengji Luo, Gianluca Ranzi, Shu Wang und Zhao Yang Dong. „Hierarchical Energy Management System for Home Microgrids“. In: *IEEE Transactions on Smart Grid* 10.5 (Sep. 2019), Seiten 5536–5546. ISSN: 1949-3053. DOI: 10.1109/TSG.2018.2884323.
- [Lut+16] Christian Lutz, Markus Flaute, Anett Großmann und Christian Lutz. *Gesamtwirtschaftliche Effekte von Prosumer-Haushalten in Deutschland*. Technischer Bericht 05. GWS Discussion Paper, 2016.
- [Mah+15] Anzar Mahmood, Nadeem Javaid, Muhammad Asghar Khan und Sohail Razzaq. „An overview of load management techniques in smart grid“. In: *International Journal of Energy Research* 39.11 (Sep. 2015), Seiten 1437–1450. ISSN: 0363907X. DOI: 10.1002/er.3350.
- [MBR15] Roberto Minerva, Abyi Biru und Domenico Rotondi. „Towards a definition of the Internet of Things (IoT)“. In: *IEEE Internet Things* (2015), Seiten 1–86. URL: [http://iot.ieee.org/images/files/pdf/IEEE\\_IoT\\_Towards\\_Definition\\_Internet\\_of\\_Things\\_Revision1\\_27MAY15.pdf](http://iot.ieee.org/images/files/pdf/IEEE_IoT_Towards_Definition_Internet_of_Things_Revision1_27MAY15.pdf).
- [MFG15] Omid Motlagh, Greg Foliente und George Grozev. „Knowledge-Mining the Australian Smart Grid Smart City Data: A Statistical-Neural Approach to Demand-Response Analysis“. In: *Lecture Notes in Geoinformation and Cartography*. Band 213. 2015, Seiten 189–207. ISBN: 9783319183671. DOI: 10.1007/978-3-319-18368-8\_10.
- [MG11] P M Mell und Timothy Grance. *The NIST definition of cloud computing*. Technischer Bericht 800-145. Gaithersburg, MD: National Institute of Standards und Technology, 2011. DOI: 10.6028/NIST.SP.800-145.
- [Mod18] Modbus Organization. *MODBUS/TCP Security, Protocol Specification*. 2018.



- 
- [Mon15] Andres Monzon. „Smart cities concept and challenges: Bases for the assessment of smart city projects“. In: *2015 International Conference on Smart Cities and Green ICT Systems (SMARTGREENS)*. 2015. URL: <https://ieeexplore.ieee.org/abstract/document/7297938>.
- [MS06] Peter Mahlmann und Christian Schindelhauer. *Peer-to-Peer-Netzwerke*. eXamen.press. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. ISBN: 978-3-540-33991-5. DOI: 10.1007/978-3-540-33992-2.
- [NDM18] C. Nayanatara, S. Divya und E.K. Mahalakshmi. „Micro-Grid Management Strategy with the Integration of Renewable Energy Using IoT“. In: *2018 Internat2018 International Conference on Computation of Power, Energy, Information and Communication (ICCPEIC)*ional conference on computation of power, energy, Information and Communication (ICCPEIC). IEEE, März 2018, Seiten 160–165. ISBN: 978-1-5386-2447-0. DOI: 10.1109/ICCPEIC.2018.8525205.
- [PD11] Peter Palensky und Dietmar Dietrich. „Demand Side Management: Demand Response, Intelligent Energy Systems, and Smart Loads“. In: *IEEE Transactions on Industrial Informatics* 7.3 (Aug. 2011), Seiten 381–388. ISSN: 1551-3203. DOI: 10.1109/TII.2011.2158841.
- [PG18] K. T. Prajwal und V S N V Sitaram Gupta. „Smart Home Energy Management System using Fuzzy Logic for Continuous Power Supply with Economic Utilisation of Electrical Energy“. In: *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. ICISC. IEEE, Jan. 2018, Seiten 274–279. ISBN: 978-1-5386-0807-4. DOI: 10.1109/ICISC.2018.8399078.
- [Pil+18] Virginia Pilloni, Alessandro Floris, Alessio Meloni und Luigi Atzori. „Smart Home Energy Management Including Renewable Sources: A QoE-driven Approach“. In: *IEEE Transactions on Smart Grid* 9.3 (2018). ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2605182.
- [PK08] Beat Pfister und Tobias Kaufmann. *Sprachverarbeitung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, Seite 43. ISBN: 978-3-540-75909-6. DOI: 10.1007/978-3-540-75910-2.
- [PLT16] Cu Pham, Yuto Lim und Yasuo Tan. „Management architecture for heterogeneous IoT devices in home network“. In: *2016 IEEE 5th Global Conference on Consumer Electronics*. IEEE, Okt. 2016, Seiten 1–5. ISBN: 978-1-5090-2333-2. DOI: 10.1109/GCCE.2016.7800448.

- [Pos16] Gerrit Posselt. *Towards Energy Transparent Factories*. Sustainable Production, Life Cycle Engineering and Management. Cham: Springer International Publishing, 2016, Seite 284. ISBN: 978-3-319-20868-8. DOI: 10.1007/978-3-319-20869-5.
- [PRO03] P. Palensky, P. Rössler und D. Dietrich O. „Heim und Gebäudeautomatisierung zur Effizienzsteigerung in Gebäuden“. In: *e&i, elektrotechnik und informationstechnik* 120.4 (2003), Seiten 104–110. URL: <http://link.springer.com/article/10.1007/BF03053910>.
- [PW16] Poornima Padmanabhan und Gary R Waissi. „Cloud-based Home Energy Management (HEM) and Modeling of Consumer Decisions“. In: *International Journal of Smart Home* 10.8 (Aug. 2016), Seiten 213–232. ISSN: 19754094. DOI: 10.14257/ijsh.2016.10.8.21.
- [Rab89] L.R. Rabiner. „A tutorial on hidden Markov models and selected applications in speech recognition“. In: *Proceedings of the IEEE* 77.2 (1989), Seiten 257–286. ISSN: 00189219. DOI: 10.1109/5.18626.
- [Rah+16] Sahar Rahim, Nadeem Javaid, Ashfaq Ahmad u. a. „Exploiting heuristic algorithms to efficiently utilize energy management controllers with renewable energy sources“. In: *Energy and Buildings* 129 (Okt. 2016), Seiten 452–470. ISSN: 03787788. DOI: 10.1016/j.enbuild.2016.08.008.
- [Ray18] P.P. Ray. „A survey on Internet of Things architectures“. In: *Journal of King Saud University - Computer and Information Sciences* 30.3 (Juli 2018), Seiten 291–319. ISSN: 13191578. DOI: 10.1016/j.jksuci.2016.10.003.
- [Raz+16] Mohammad Abdur Razzaque, Marija Milojevic-Jevric, Andrei Palade und Siobhan Clarke. „Middleware for Internet of Things: A Survey“. In: *IEEE Internet of Things Journal* 3.1 (Feb. 2016), Seiten 70–95. ISSN: 2327-4662. DOI: 10.1109/JIOT.2015.2498900.
- [Reb02] Eckhard Rebhan. *Energiehandbuch*. Herausgegeben von Eckhard Rebhan. Berlin / Heidelberg: Springer Berlin Heidelberg, 2002. ISBN: 978-3-642-62518-3. DOI: 10.1007/978-3-642-55451-3.
- [Rei+12] Andreas Reinhardt, Paul Baumann, Daniel Burgstahler u. a. „On the Accuracy of Appliance Identification Based on Distributed Load Metering Data“. In: *Proceedings of the 2nd IFIP Conference on Sustainable Internet and ICT for Sustainability* October (2012), Seiten 1–9. URL: <https://ieeexplore.ieee.org/abstract/document/6388037>.

- 
- [Rod+14] Jesús Rodríguez-Molina, Margarita Martínez-Núñez, José-Fernán Martínez und Waldo Pérez-Aguilar. „Business Models in the Smart Grid: Challenges, Opportunities and Proposals for Prosumer Profitability“. In: *Energies* 7.9 (Sep. 2014), Seiten 6142–6171. ISSN: 1996-1073. DOI: 10.3390/en7096142.
- [RSS15] D Romascanu, J. Schoenwaelder und A Sehgal. *Management of Networks with Constrained Devices: Use Cases*. Herausgegeben von M. Ersue. Mai 2015. DOI: 10.17487/RFC7548.
- [Sca15] Scalagent. *Benchmark of MQTT servers*. Technischer Bericht January. 2015, Seiten 1–15. URL: [http://www.scalagent.com/IMG/pdf/Benchmark\\_MQTT\\_servers-v1-1.pdf](http://www.scalagent.com/IMG/pdf/Benchmark_MQTT_servers-v1-1.pdf).
- [Sch+15] Marc Schickler, Manfred Reichert, Rüdiger Pryss u. a. „Arten der Anwendungsentwicklung“. In: *Entwicklung mobiler Apps: Konzepte, Anwendungsbausteine und Werkzeuge im Business und E-Health*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, Seiten 17–24. ISBN: 978-3-642-33057-5. DOI: 10.1007/978-3-642-33057-5\_4.
- [Sch05] Peter Scholz. *Softwareentwicklung eingebetteter Systeme*. Band 6. Xpert.press 2. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, Seite 103. ISBN: 978-3-540-23405-0. DOI: 10.1007/3-540-27522-3.
- [Sch18] Miranda Schreurs. „Gemeinsam lernen – Internationaler Erfahrungsaustausch in der Energiewende. Der Fall Deutschland und Japan“. In: *Handbuch Energiewende und Partizipation*. Wiesbaden: Springer Fachmedien Wiesbaden, 2018, Seiten 157–174. ISBN: 9783658094164. DOI: 10.1007/978-3-658-09416-4\_10.
- [SCN15] B. Schoening, M. Chandramouli und B. Nordman. *Energy Management (EMAN) Applicability Statement*. Aug. 2015. DOI: 10.17487/RFC7603.
- [Sha+18] Hussain Shareef, Maytham S. Ahmed, Azah Mohamed und Eslam Al Hassan. „Review on Home Energy Management System Considering Demand Responses, Smart Technologies, and Intelligent Controllers“. In: *IEEE Access* 6 (2018), Seiten 24498–24509. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2018.2831917.
- [Sin10] Otto Singer. *Aktueller Begriff - Cloud Computing*. Technischer Bericht 15. 2010, Seite 2. URL: [http://www.bundestag.de/blob/191178/22a7553089d81c2e06866e15fc354a0e/cloud\\_computing-data.pdf](http://www.bundestag.de/blob/191178/22a7553089d81c2e06866e15fc354a0e/cloud_computing-data.pdf).
- [Sou+18] Alisson Trindade Souza, Rafael Gressler Milbradt, Luciane Neves Canha und Tiago Augusto Silva Santana. „Development of a Demand Response System Integrated to Photovoltaic Microgeneration and Energy Storage Using IoT and Artificial Intelligence“. In: *2018 15th International Conference on the European*

- Energy Market (EEM)*. Band 2018-June. IEEE, Juni 2018, Seiten 1–5. ISBN: 978-1-5386-1488-4. DOI: 10.1109/EEM.2018.8469971.
- [Sow+18] Robert A. Sowah, Abdul R. Ofoli, Michael K. Tetteh u. a. „Demand Side Management of Smart Homes Using OpenHAB Framework for Interoperability of Devices“. In: *2018 IEEE 7th International Conference on Adaptive Science & Technology (ICAST)*. Band 2018-Augus. IEEE, Aug. 2018, Seiten 1–8. ISBN: 978-1-5386-4233-7. DOI: 10.1109/ICASTECH.2018.8506917.
- [Spo+14] Manu Sporny, Digital Bazaar, Gregg Kellogg u. a. *JSON-LD 1.0 – A JSON-based Serialization for Linked Data*. Recommendation. W3C, Jan. 2014. URL: <https://www.w3.org/TR/2014/REC-json-ld-20140116/>.
- [ST17] Maarten van Steen und Andrew S. Tanenbaum. *Distributed Systems*. 3.01. CreateSpace Independent Publishing Platform, 2017, Seite 596. ISBN: 978-1543057386. URL: <https://www.distributed-systems.net>.
- [Sta14] John A Stankovic. „Research Directions for the Internet of Things“. In: *IEEE Internet of Things Journal* 1.1 (Feb. 2014), Seiten 3–9. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2312291.
- [Tas+15] Akin Tascikaraoglu, Nikolaos G. Paterakis, Joao P. S. Catalao u. a. „An EMD-ANN based prediction methodology for DR driven smart household load demand“. In: *2015 18th International Conference on Intelligent System Application to Power Systems (ISAP)*. IEEE, Sep. 2015, Seiten 1–6. ISBN: 978-1-5090-0191-0. DOI: 10.1109/ISAP.2015.7325544.
- [TV10] Stefan Tilkov und Steve Vinoski. „Node.js: Using JavaScript to Build High-Performance Network Programs“. In: *IEEE Internet Computing* 14.6 (Nov. 2010), Seiten 80–83. ISSN: 1089-7801. DOI: 10.1109/MIC.2010.145.
- [Ver07] Verband der Netzbetreiber (VDN). *TransmissionCode 2007*. Technischer Bericht. Berlin, Aug. 2007, Seiten 1–90. URL: <https://www.regelleistung.net/ext/download/transmission2007%0A%0A>.
- [Vij+18] V. Vijeesh, B Ponkarthika, E. Kaliappan u. a. „Energy Saving and Smart Billing System for Household Consumers Connected to a Smart Grid in Tamil Nadu Power System“. In: *2018 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*. IEEE, Feb. 2018, Seiten 233–239. ISBN: 978-1-5386-3817-0. DOI: 10.1109/ICPECTS.2018.8521613.

- 
- [Wan+18] Xiaojie Wang, Zhaolong Ning, MengChu Zhou u. a. „A Privacy-Preserving Message Forwarding Framework for Opportunistic Cloud of Things“. In: *IEEE Internet of Things Journal* 5.6 (Dez. 2018), Seiten 5281–5295. DOI: 10.1109/JIOT.2018.2864782.
- [WLR18] Fernando Marcos Wittmann, Juan Camilo Lopez und Marcos J. Rider. „Nonintrusive Load Monitoring Algorithm Using Mixed-Integer Linear Programming“. In: *IEEE Transactions on Consumer Electronics* 64.2 (Mai 2018), Seiten 180–187. ISSN: 0098-3063. DOI: 10.1109/TCE.2018.2843292.
- [WSJ15] Roy Want, Bill N. Schilit und Scott Jenson. „Enabling the Internet of Things“. In: *Computer* 48.1 (Jan. 2015), Seiten 28–35. ISSN: 0018-9162. DOI: 10.1109/MC.2015.12.
- [Wu+18] Xiaohua Wu, Xiaosong Hu, Xiaofeng Yin und Scott J. Moura. „Stochastic Optimal Energy Management of Smart Home With PEV Energy Storage“. In: *IEEE Transactions on Smart Grid* 9.3 (Mai 2018), Seiten 2065–2075. ISSN: 1949-3053. DOI: 10.1109/TSG.2016.2606442.
- [Xia+18] Chunqiu Xia, Wei Li, Xiaomin Chang u. a. „Edge-based Energy Management for Smart Homes“. In: *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*. IEEE, Aug. 2018, Seiten 849–856. ISBN: 978-1-5386-7518-2. DOI: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00–19.
- [Xie+16] Qiangqiang Xie, Ryoichi Hara, Hiroyuki Kita und Eiichi Tanaka. „A real-time cooperative overvoltage prevention control by a community energy management system“. In: *Journal of International Council on Electrical Engineering* 6.1 (2016), Seiten 224–230. ISSN: 2234-8972. DOI: 10.1080/22348972.2016.1248609.
- [Yan16] Bin Yang. „Nonintrusive Load Monitoring (NILM). From unsupervised source separation to supervised source extraction“. In: *NILM Workshop 2016*. London, 2016. URL: <http://www.nilmm.eu/nilm-workshop-2016/>.
- [YHS18] M. Yousefi, A. Hajizadeh und M. Soltani. „Energy management strategies for smart home regarding uncertainties: State of the art, trends, and challenges“. In: *2018 IEEE International Conference on Industrial Technology (ICIT)*. IEEE, Feb. 2018, Seiten 1219–1225. ISBN: 978-1-5090-5949-2. DOI: 10.1109/ICIT.2018.8352352.

- [ZC18] Azer Zairi und Maher Chaabene. „A review on home energy management systems“. In: *2018 9th International Renewable Energy Congress (IREC)*. Irec. IEEE, März 2018, Seiten 1–6. ISBN: 978-1-5386-0998-9. DOI: 10.1109/IREC.2018.8362475.
- [Zha+19] Shaopeng Zhai, Zhihua Wang, Xufeng Yan und Guangyu He. „Appliance Flexibility Analysis Considering User Behavior in Home Energy Management System Using Smart Plugs“. In: *IEEE Transactions on Industrial Electronics* 66.2 (Feb. 2019), Seiten 1391–1401. ISSN: 0278-0046. DOI: 10.1109/TIE.2018.2815949.
- [Zha18] Dongpo Zhang. „Big Data Security and Privacy Protection“. In: *Proceedings of the 8th International Conference on Management and Computer Science (ICMCS 2018)*. Paris, France: Atlantis Press, 2018, Seiten 275–278. ISBN: 978-94-6252-590-0. DOI: 10.2991/icmcs-18.2018.56.
- [Zho+16] Bin Zhou, Wentao Li, Ka Wing Chan u. a. „Smart home energy management systems: Concept, configurations, and scheduling strategies“. In: *Renewable and Sustainable Energy Reviews* 61 (Aug. 2016), Seiten 30–40. ISSN: 13640321. DOI: 10.1016/j.rser.2016.03.047.
- [Zoh+13] Ahmed Zoha, Alexander Gluhak, Michele Nati und Muhammad Ali Imran. „Low-power appliance monitoring using Factorial Hidden Markov Models“. In: *2013 IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*. IEEE, Apr. 2013, Seiten 527–532. ISBN: 978-1-4673-5501-8. DOI: 10.1109/ISSNIP.2013.6529845.